

SWORD

XD-LAB-ARCH-004

Lab4: Pipelined CPU resolving control hazard

Hawk

2019-1-19

XINGDENG

修改记录

版本号.	作者	描述	修改日期
1.0	Hawk	初稿	2019-01-19

审核记录

姓名	职务	签字	日期

	标题	文档编号	版本	页
	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	1 of 34
作者	修改日期			
Hawk	2019/1/19			公开

目录

修改记录	0
审核记录	1
1. 实验准备	5
1. 1 实验目的	5
1. 2 实验内容	5
1. 3 实验资料	5
2. 实验步骤	7
2. 1 新建工程	7
2. 2 修改文件	9
2. 3 功能仿真	24
2. 4 生成 bitstream 文件	28
2. 5 下载验证	30

标题	文档编号	版本	页
XINGDENG Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	2 of 34
作者 Hawk	修改日期 2019/1/19		公开

图目录

图 1-1 实验资料网站	5
图 1-2 实验文档和工程源文件	6
图 2-1 运行 Vivado	7
图 2-2 打开工程	7
图 2-3 打开 Lab03 工程	8
图 2-4 工程另存为	8
图 2-5 工程另存为 Lab04	9
图 2-6 需要修改的 lab04 工程	9
图 2-7 展开 mips_top	10
图 2-8 删除文件	10
图 2-9 删 除文件	11
图 2-10 继续删除	11
图 2-11 增加文件	12
图 2-12 增加 design sources	12
图 2-13 add files	13
图 2-14 切换目录	13
图 2-15 选择文件	14
图 2-16 增加文件	14
图 2-17 工程文件树	15
图 2-18 删除 hex 文件	15
图 2-19 增加 hex 文件	16
图 2-20 增加 design sources	16
图 2-21 add files	17
图 2-22 设置全部文件可见	17
图 2-23 切换目录	18
图 2-24 选中文件	18
图 2-25 设置文件类型	19
图 2-26 设置为 Data Files	20
图 2-27 更新视图	20
图 2-28 打开文件	21
图 2-29 添加信号	21

XINGDENG	标题	文档编号	版本	页
	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	3 of 34
作者	修改日期			
Hawk	2019/1/19			公开

图 2-30 修改代码.....	22
图 2-31 修改代码.....	22
图 2-32 保存修改.....	23
图 2-33 修改文件.....	23
图 2-34 保存修改.....	24
图 2-35 复制文件.....	24
图 2-36 粘贴文件.....	25
图 2-37 开始仿真.....	25
图 2-38 仿真视图.....	26
图 2-39 Run for...	26
图 2-40 缩放波形.....	27
图 2-41 仿真结果 1.....	27
图 2-42 仿真结果 2.....	28
图 2-43 仿真结果 3.....	28
图 2-44 关闭仿真.....	29
图 2-45 生成 bitstream.....	29
图 2-46 自动运行中间步骤.....	30
图 2-47 生成 bitstream 成功	30
图 2-48 连接 SWORD4.0.....	31
图 2-49 主要控制 IO	31
图 2-50 进入 Hardware Manager	32
图 2-51 自动连接到 SWORD4.0	32
图 2-52 Program Device	33
图 2-53 下载 bit 文件	33
图 2-54 运行效果.....	34

XINGDENG	标题	文档编号	版本	页
	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	4 of 34
作者	修改日期			
Hawk	2019/1/19			公开

1. 实验准备

1.1 实验目的

在实验 3 的基础上，通过 predict-not-taken 或延时转移策略解决控制竞争，减少控制竞争带来的流水线停顿延时，进一步提高处理器性能。

1.2 实验内容

本实验内容包括：

- 以 Lab03 为基础，修改设计 CPU 的数据通路 Datapath 和控制单元 Controller；
- 对设计的 CPU 进行行为仿真，检查功能正确性；
- 上板验证并观察 CPU 的执行过程。

1.3 实验资料

实验资料可以从网站 <https://gitee.com/SWORDfpga/ComputerArchitecture> 获得。

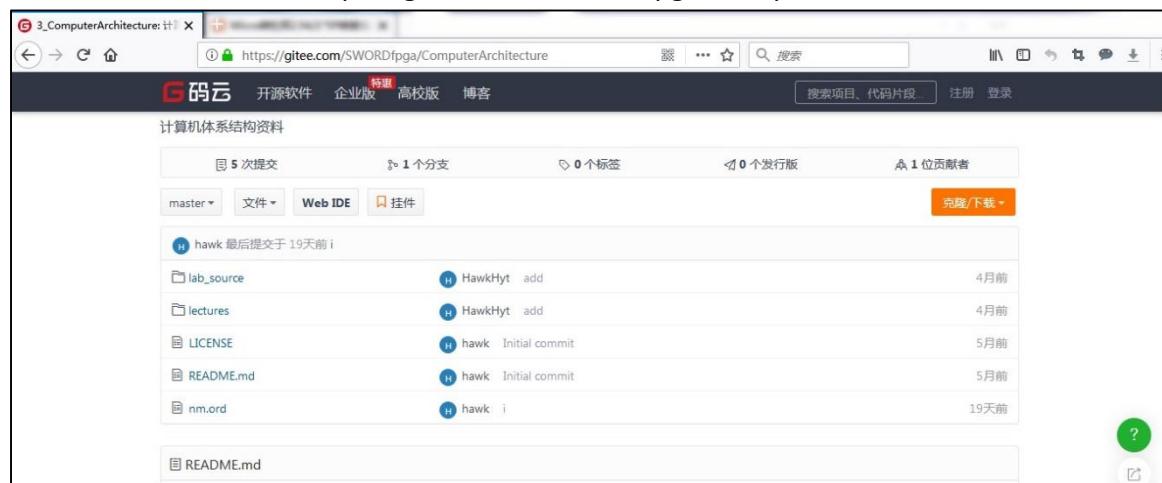


图 1-1 实验资料网站

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	5 of 34
作者 Hawk	修改日期 2019/1/19		公开

为方便使用，建议建立 c:\archlabs 目录，放置实验资料和实验工程。

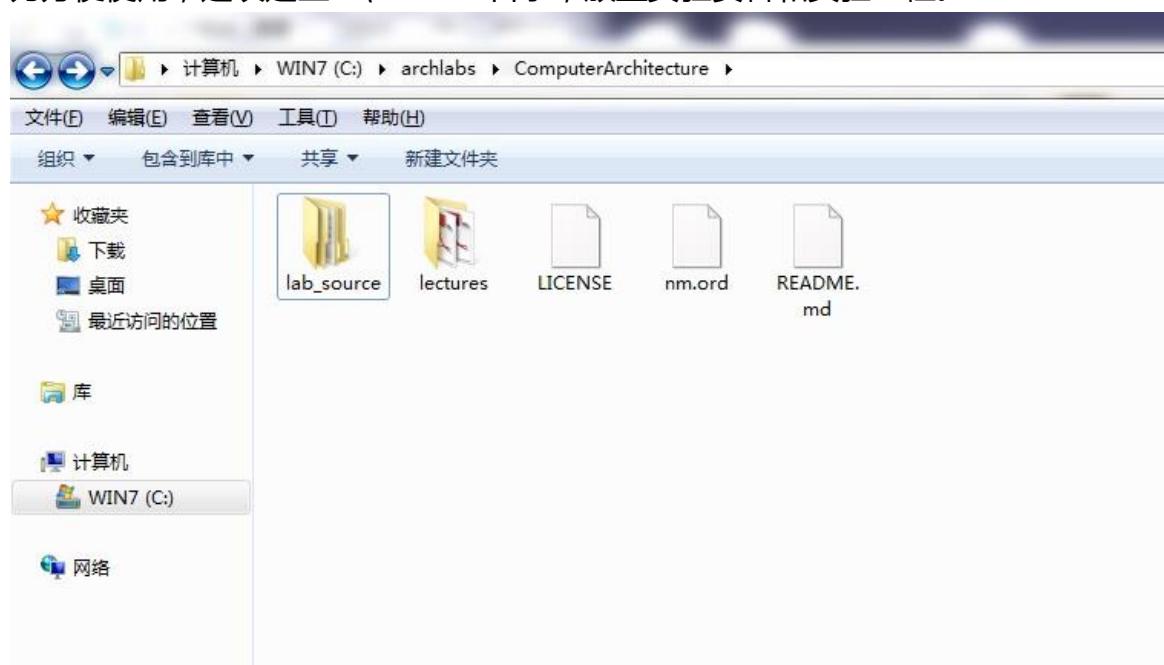


图 1-2 实验文档和工程源文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	6 of 34
作者 Hawk	修改日期 2019/1/19	公开	

2. 实验步骤

2.1 新建工程

首先启动 Vivado 2014.4 开发工具，如图 2-1 所示：



图 2-1 运行 Vivado

点击 Open Project。

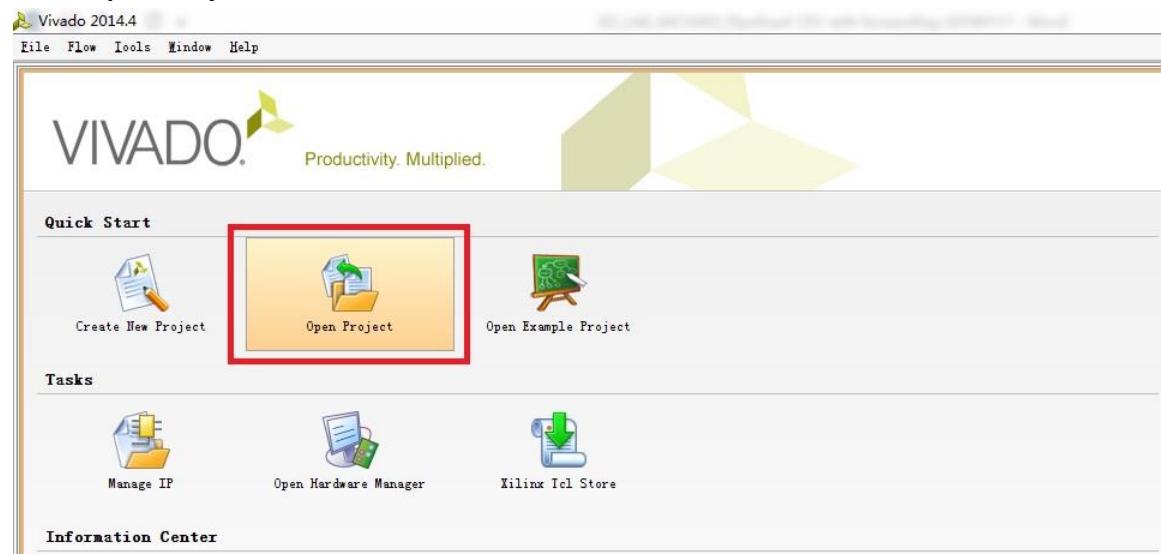


图 2-2 打开工程

XINGDENG	标题	文档编号	版本	页
	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	7 of 34
作者	修改日期			
Hawk	2019/1/19			公开

选择我们已经完成的 Lab03 工程，点击 OK 继续。

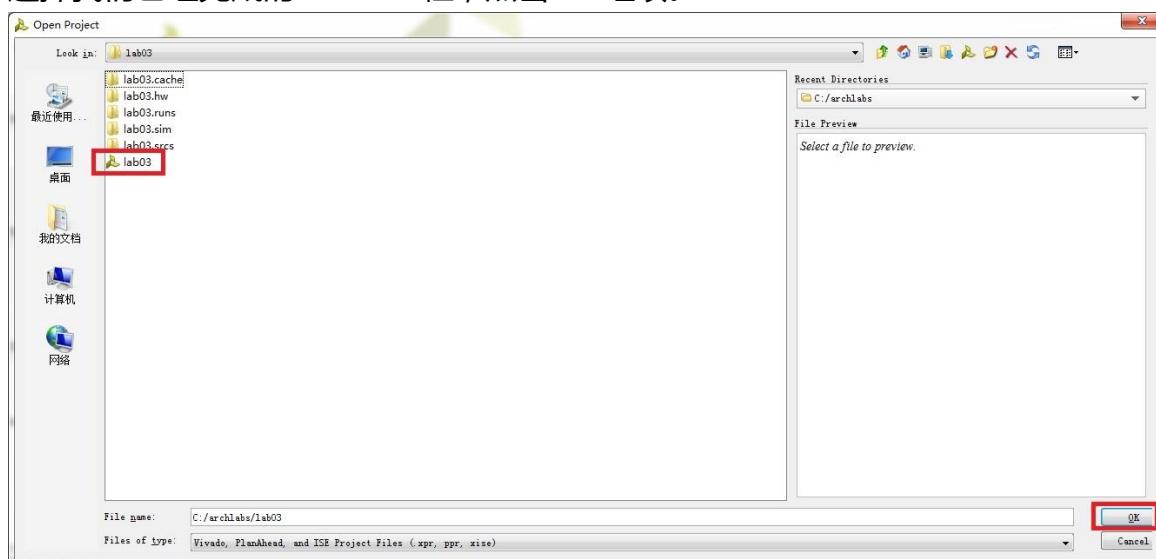


图 2-3 打开 Lab03 工程

本实验以 Lab03 工程为基础，在 File 菜单点击 Save Project As 将工程另存为。

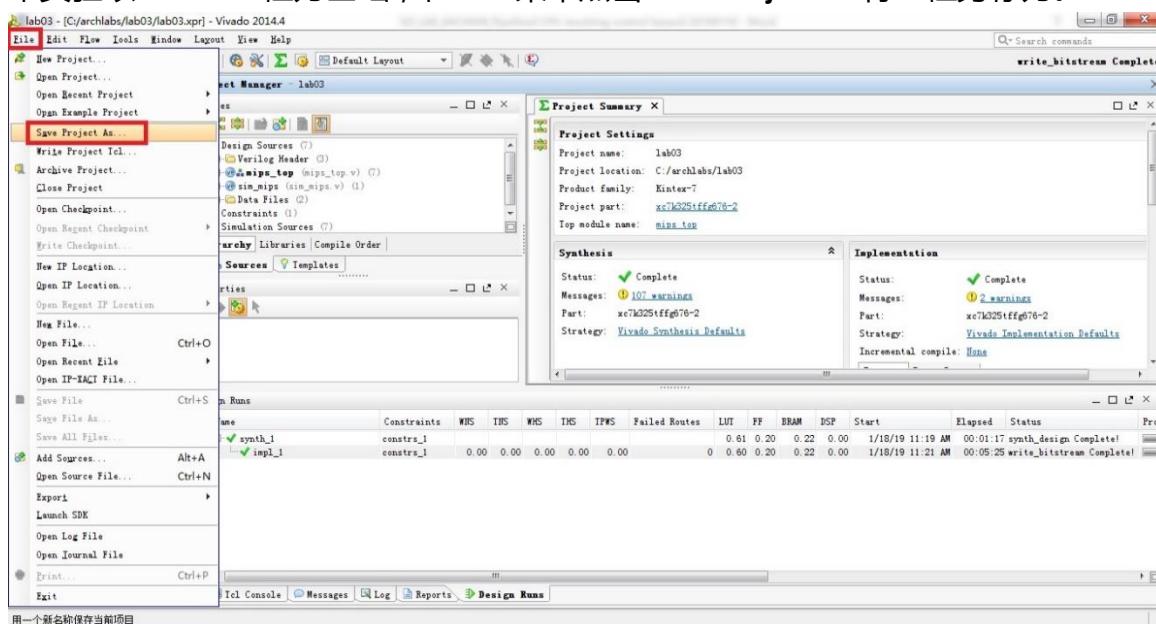


图 2-4 工程另存为

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	8 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

输入新工程名 lab04 , 选中 Create project subdirectory , 点击 OK 继续。



图 2-5 工程另存为 Lab04

工程被另存为 lab04 , 本实验将从此工程基础上进行修改。

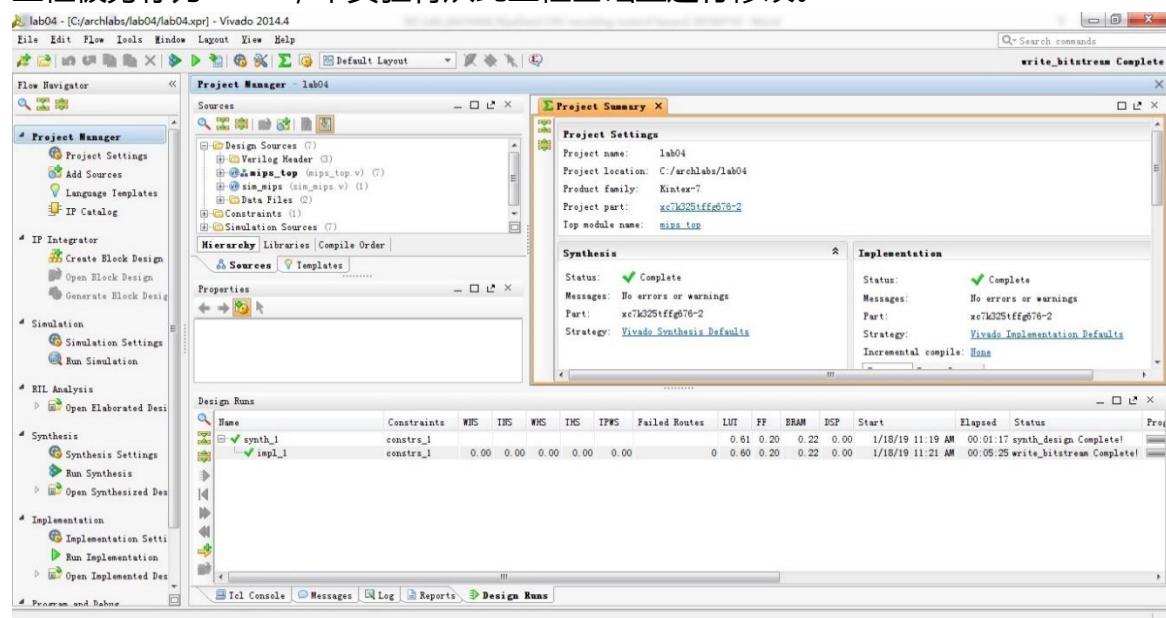


图 2-6 需要修改的 lab04 工程

2.2 修改文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	9 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

通过点击“+”展开 mips_top、MIPS 和 MIPS_CORE。

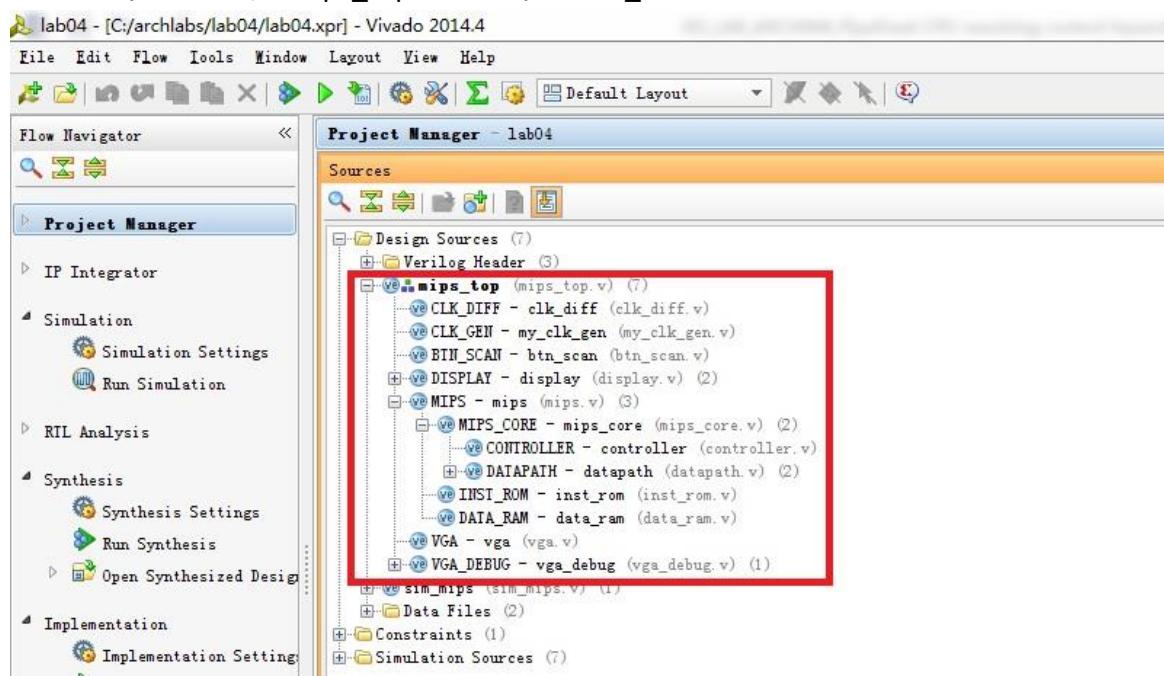


图 2-7 展开 mips_top

右键点击 DATAPATH，点击 Remove File from Project 继续。

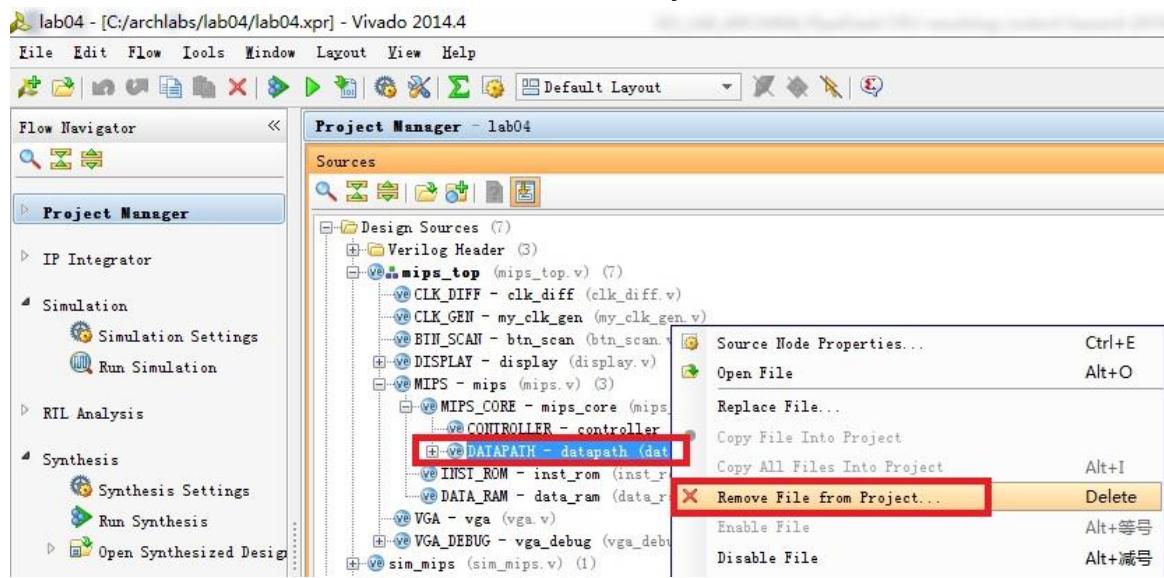


图 2-8 删除文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	10 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

选中 **Also delete the project local file/directory from disk**，点击 **OK** 继续。

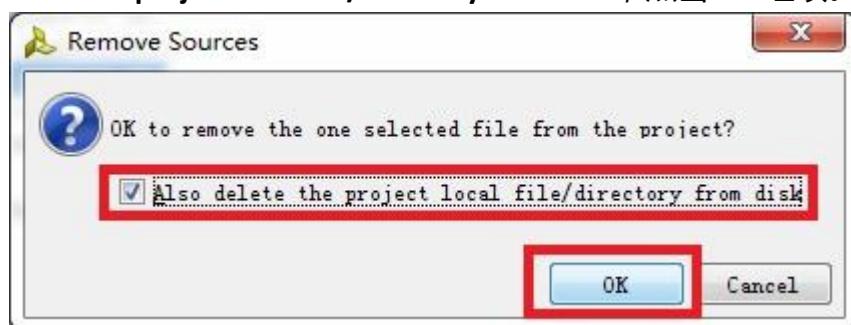


图 2-9 删除文件

继续删除文件，将 **CONTROLLER** 也通过刚才的操作删除。

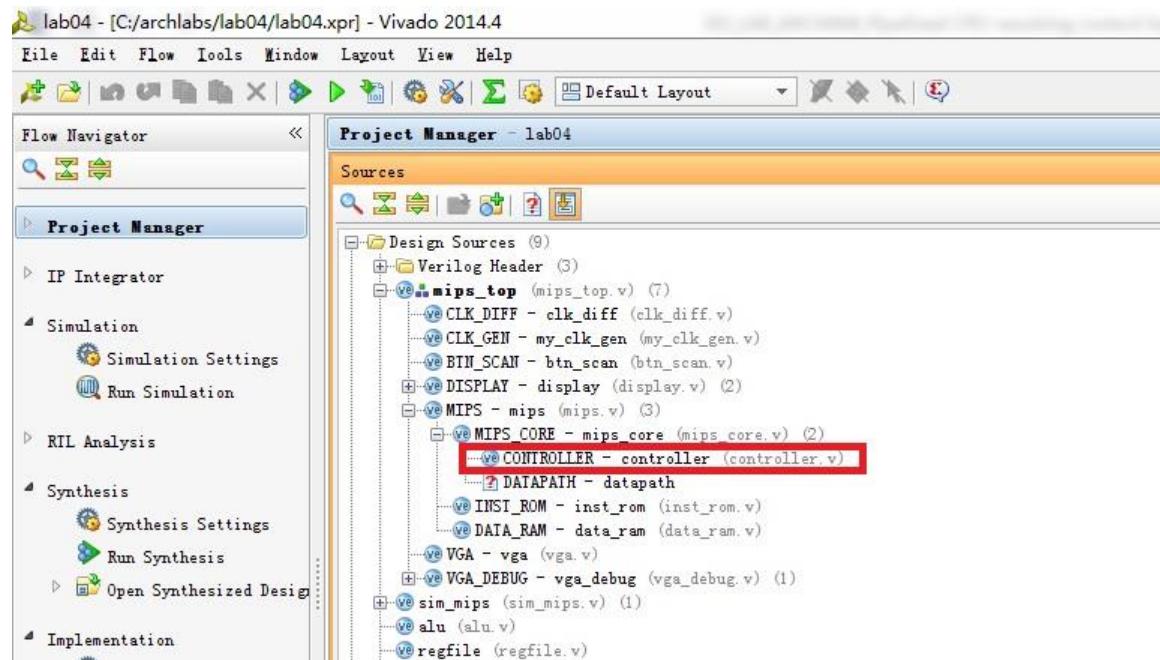


图 2-10 继续删除

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	11 of 34
作者 Hawk	修改日期 2019/1/19		公开

在 source 窗口内点击右键，点击 Add Sources。

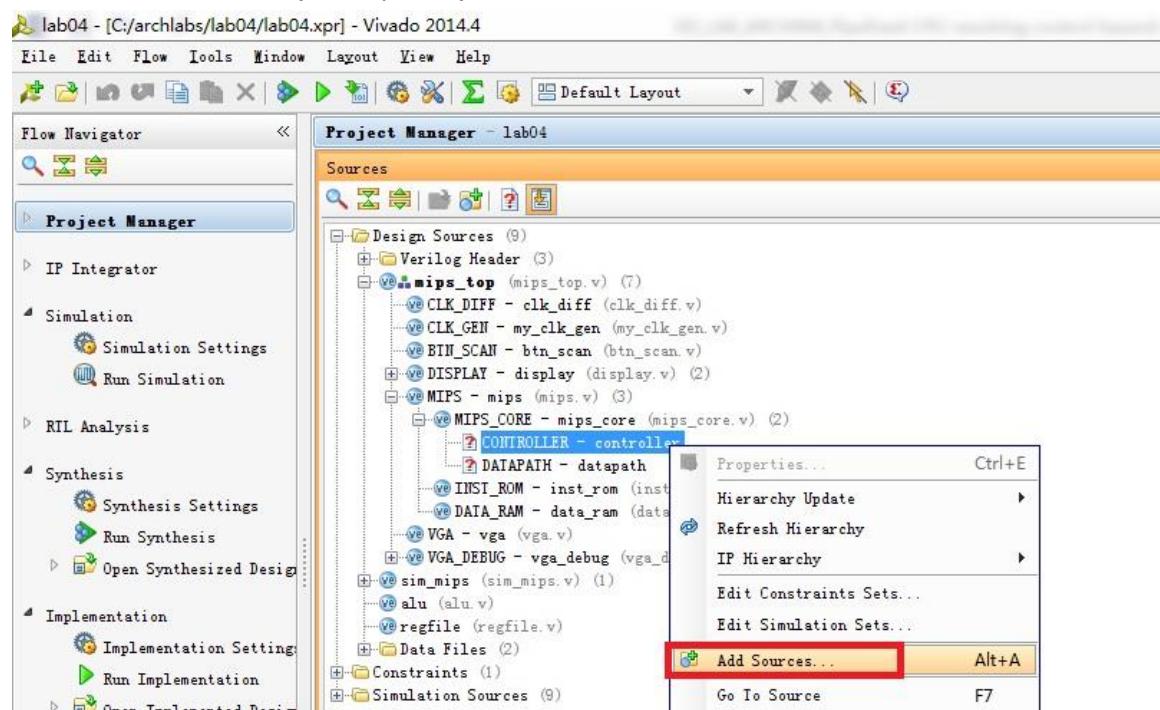


图 2-11 增加文件

选中 Add or create design sources，点击 Next。

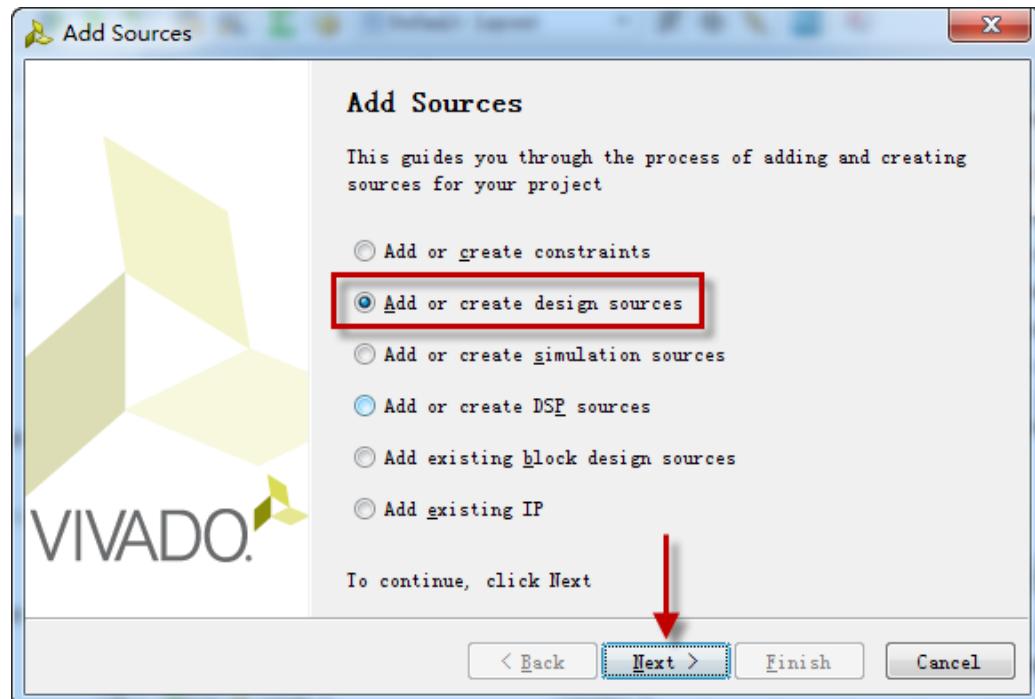


图 2-12 增加 design sources

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	12 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

点击 Add Files。

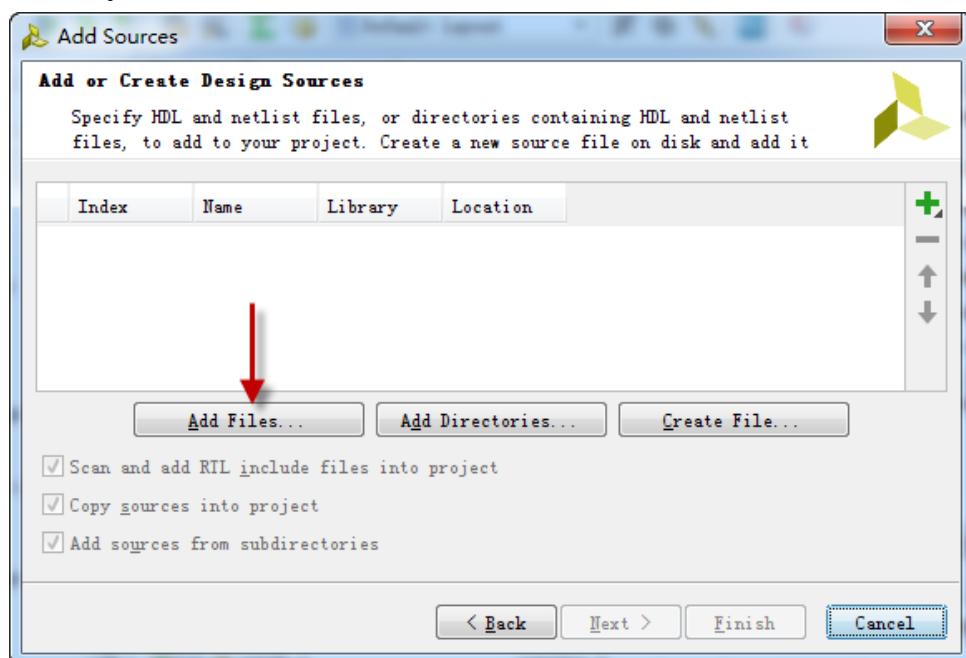


图 2-13 add files

进入 C:\archlabs\ComputerArchitecture\lab_source\lab04\sources 目录。

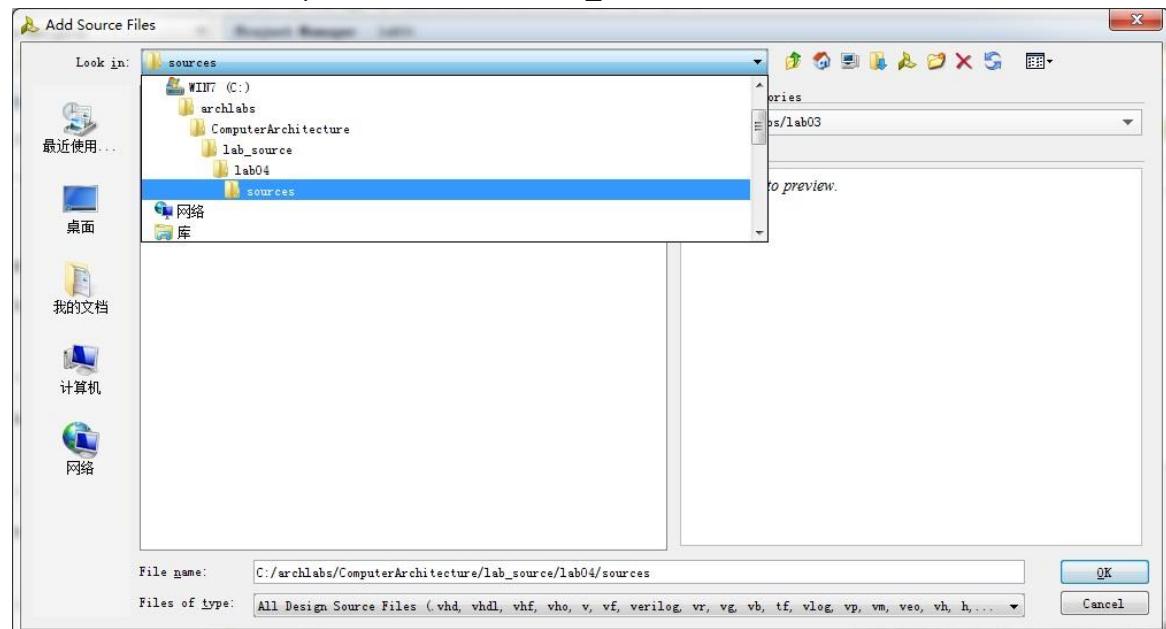


图 2-14 切换目录

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	13 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

选中所有文件，点击 OK。

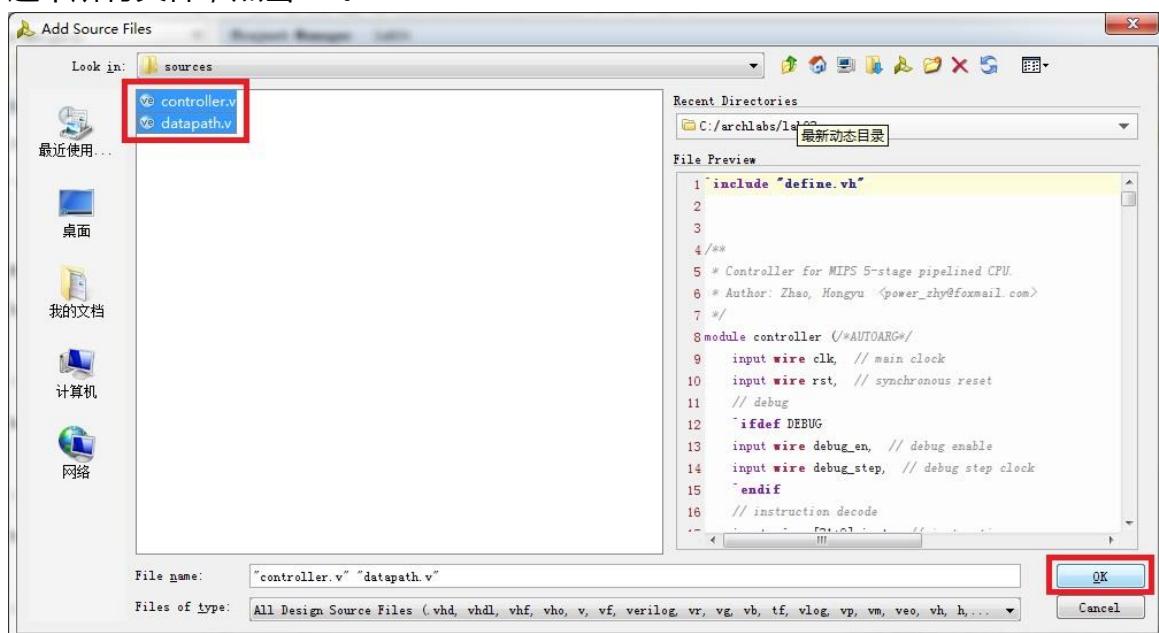


图 2-15 选择文件

列表中会显示要添加到工程中的文件，本次操作添加了 2 个文件，请确认添加文件的数量正确。选中 **Scan and add RTL include files into project** 和 **Copy sources into project**，点击 **Finish**。通过 controller.v 和 datapath.v 中的修改，减少控制竞争。

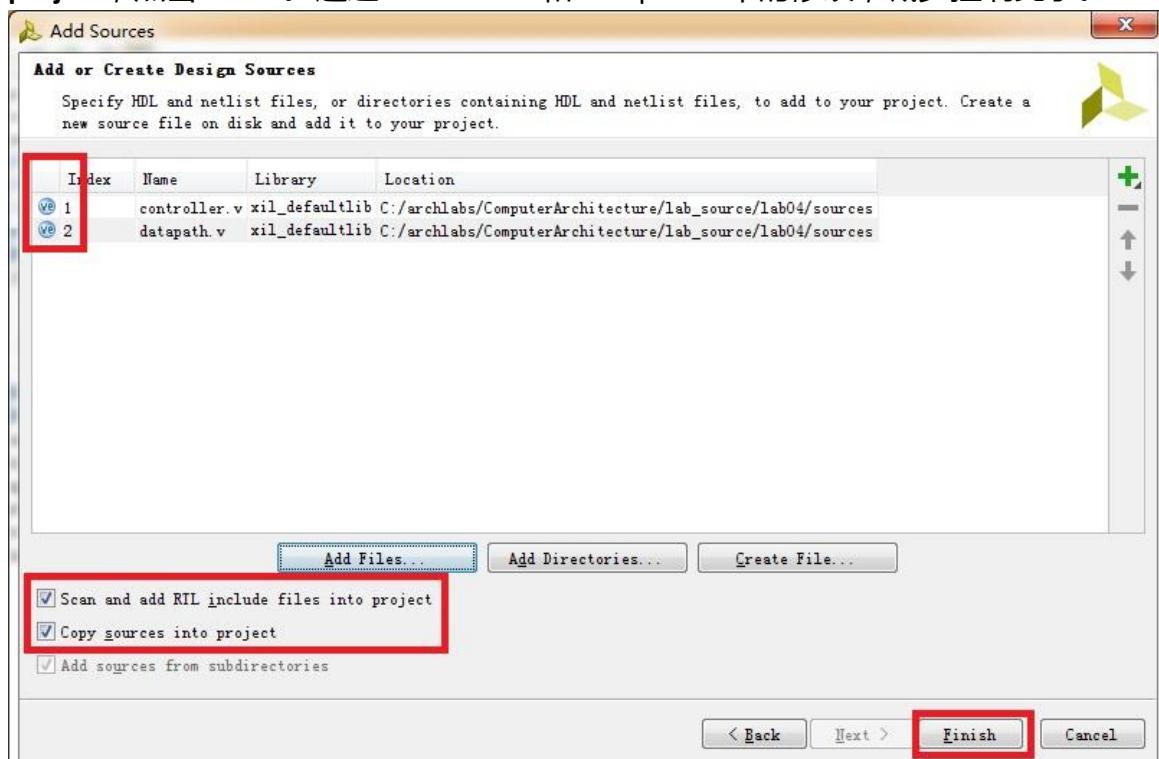


图 2-16 增加文件

标题	文档编号	版本	页
XINGDENG Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	14 of 34
作者 Hawk	修改日期 2019/1/19		公开

加入的文件替代了删除的旧文件。

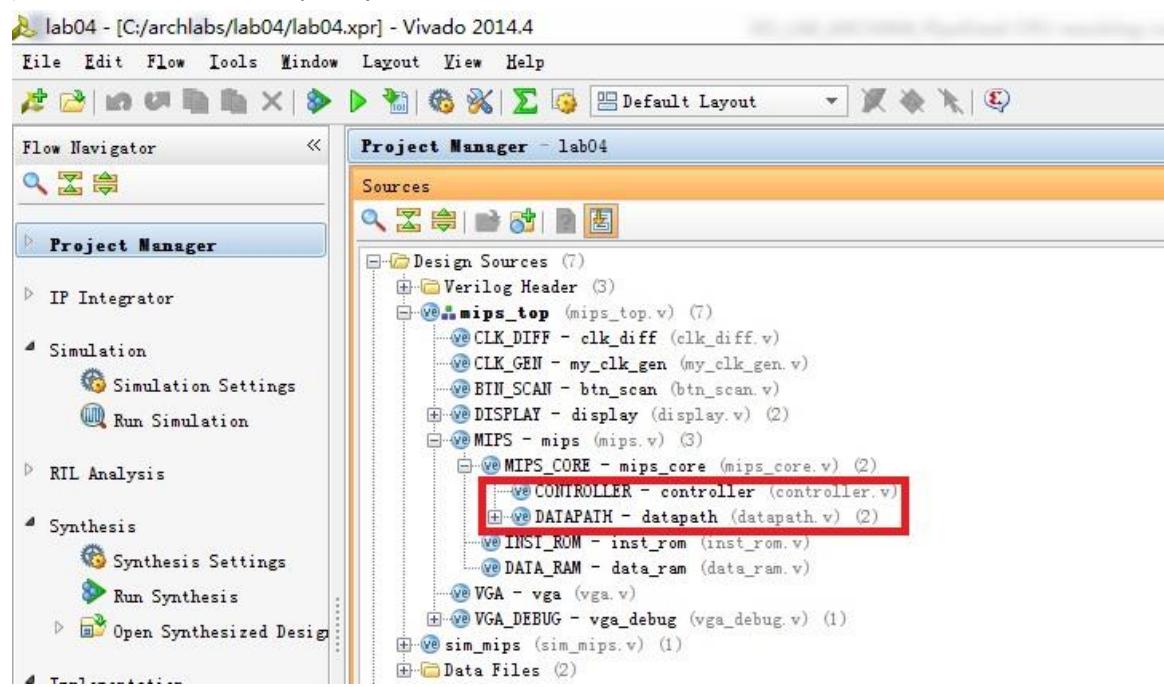


图 2-17 工程文件树

点击“+”展开 Data Files，用 Ctrl+鼠标左键选中 data_mem.hex 和 inst_mem.hex，
点击右键，点击 Remove File from Project。

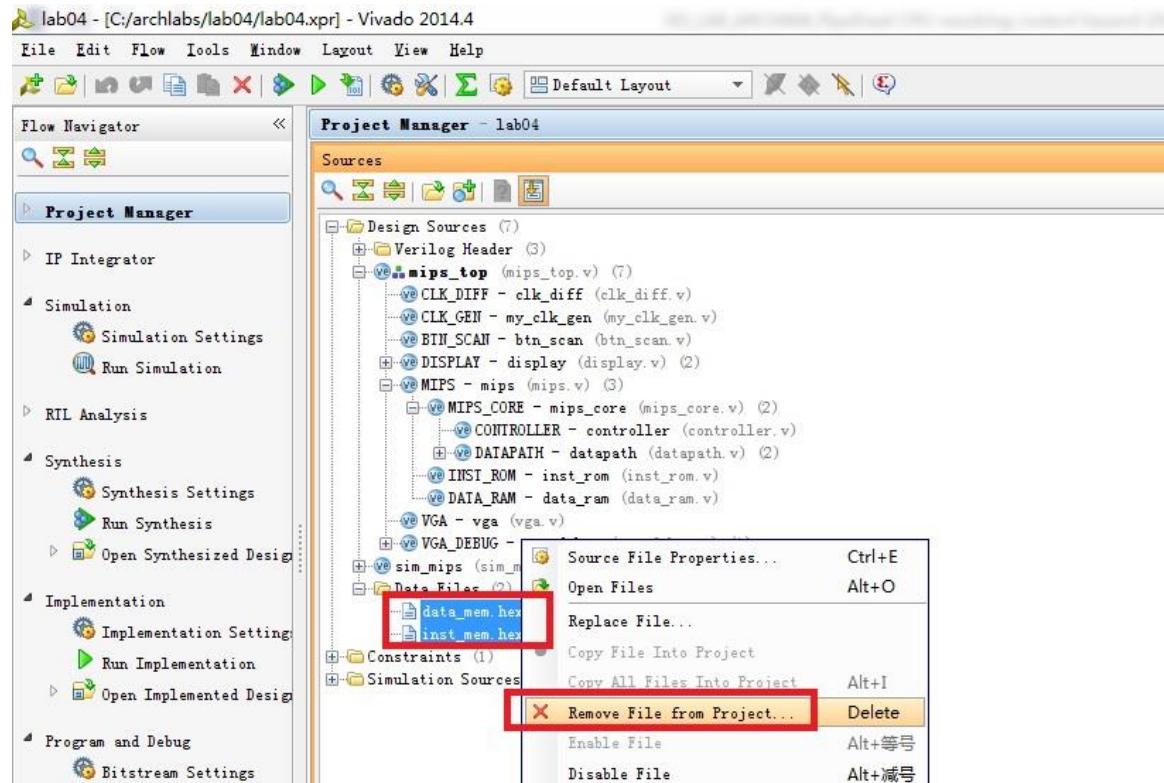


图 2-18 删除 hex 文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	15 of 34
作者 Hawk	修改日期 2019/1/19	公开	

在 Sources 窗口点击右键，点击 Add Sources，增加用于 Lab04 的 hex 文件。

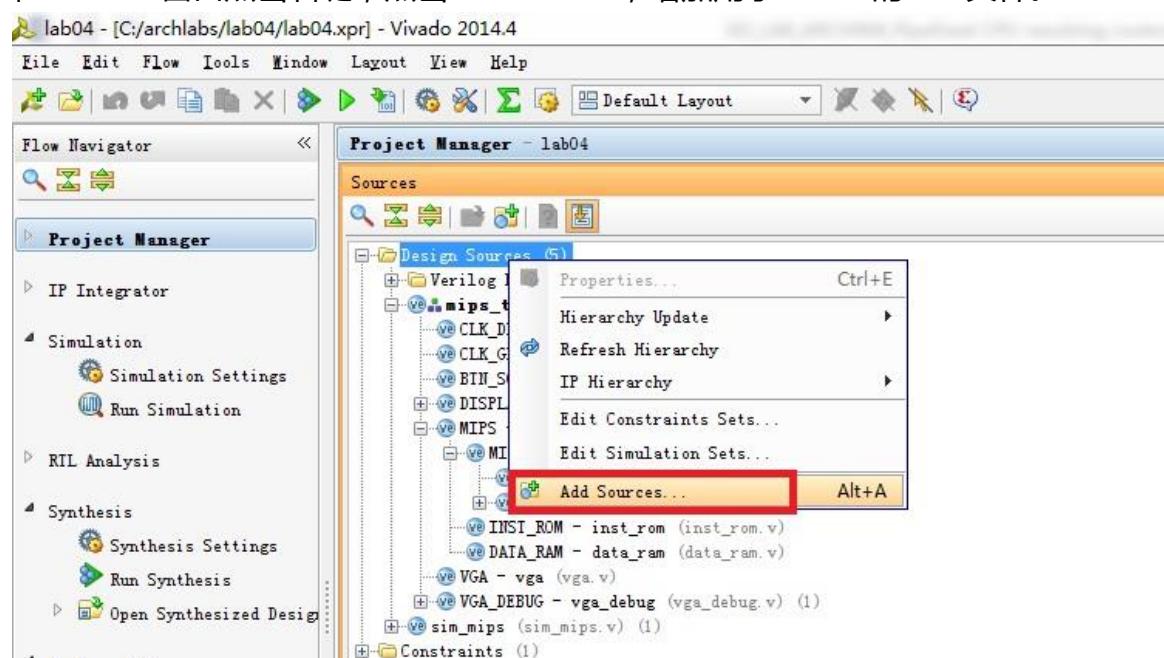


图 2-19 增加 hex 文件

选中 Add or create design sources，点击 Next。

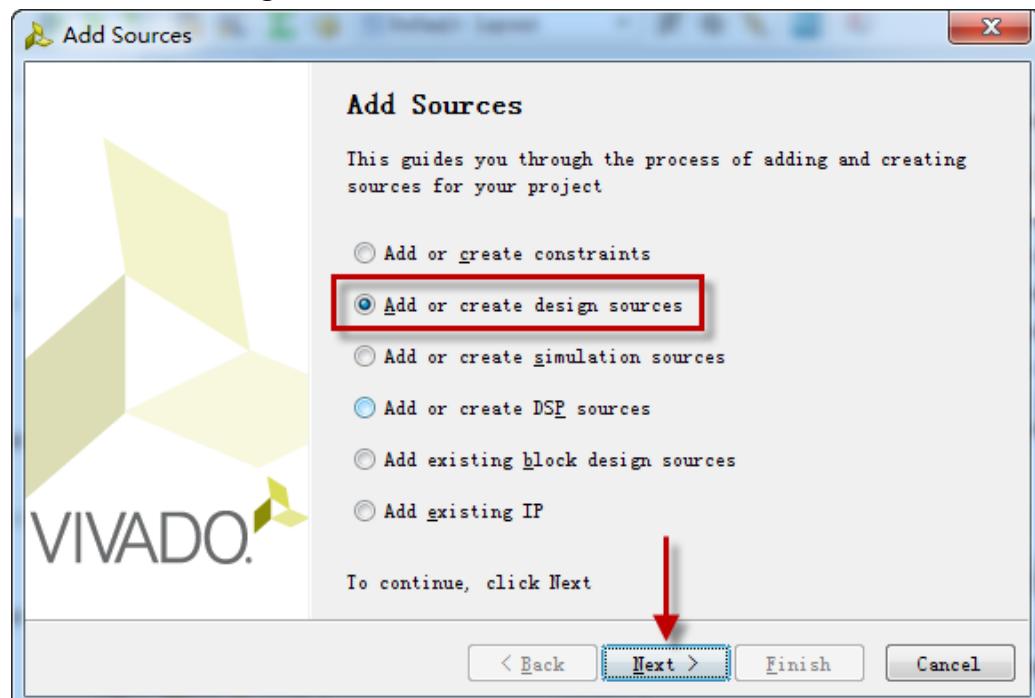


图 2-20 增加 design sources

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	16 of 34
作者 Hawk	修改日期 2019/1/19		公开

点击 Add Files。

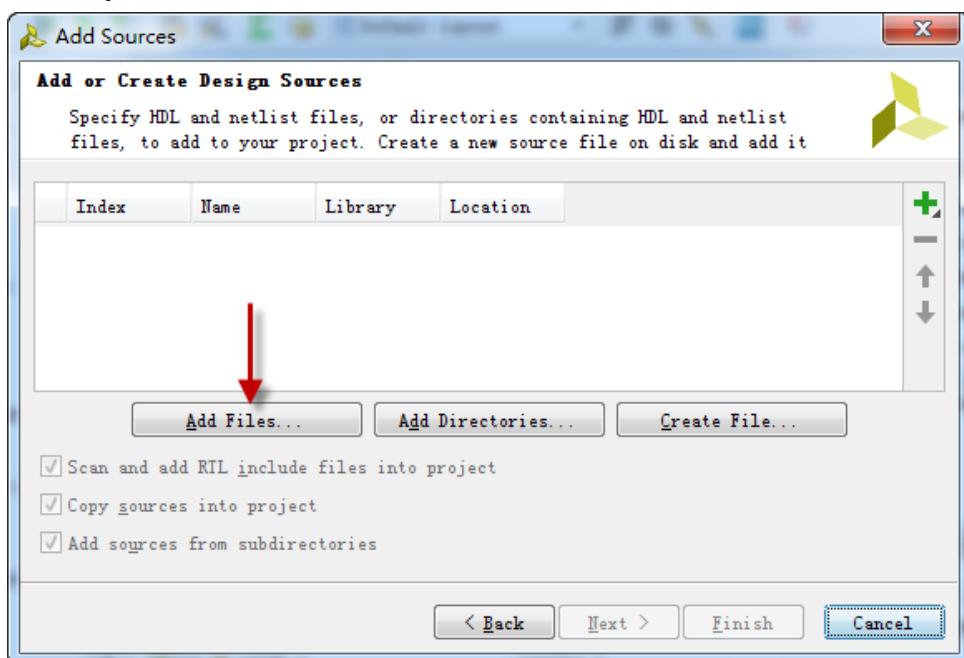


图 2-21 add files

设置 Files of type 为 All Files。

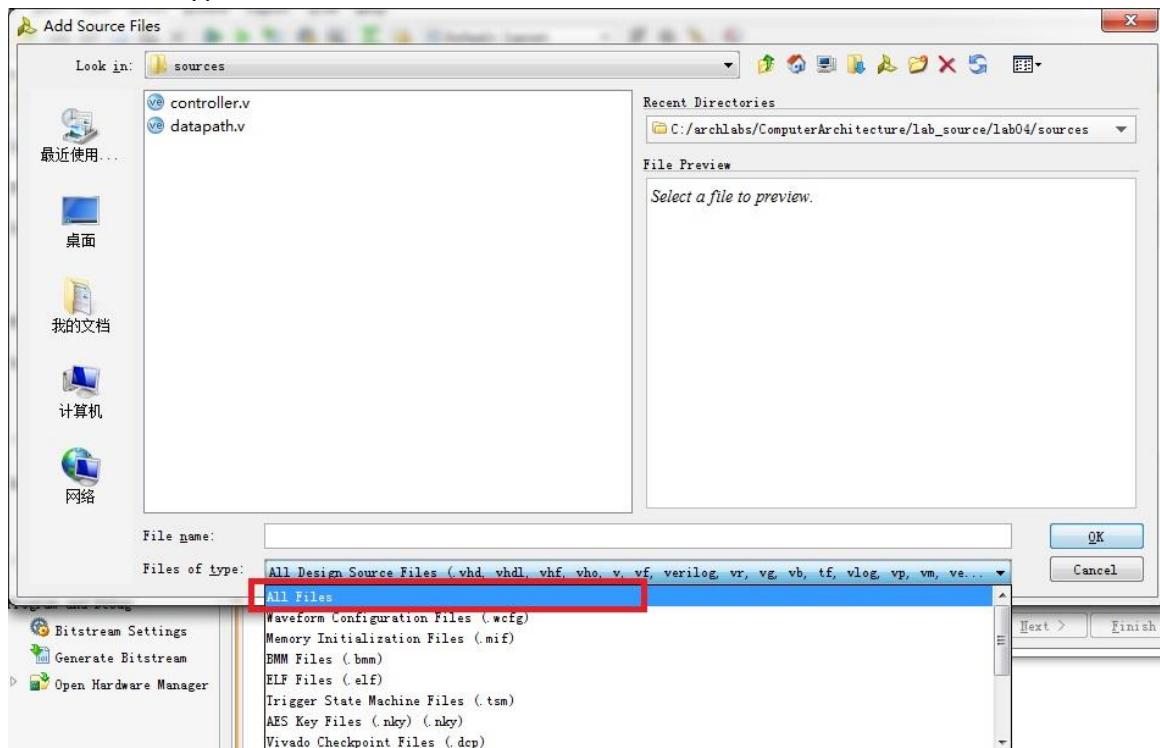


图 2-22 设置全部文件可见

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	17 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

进入 C:\archlabs\ComputerArchitecture\lab_source\lab04\data 目录。

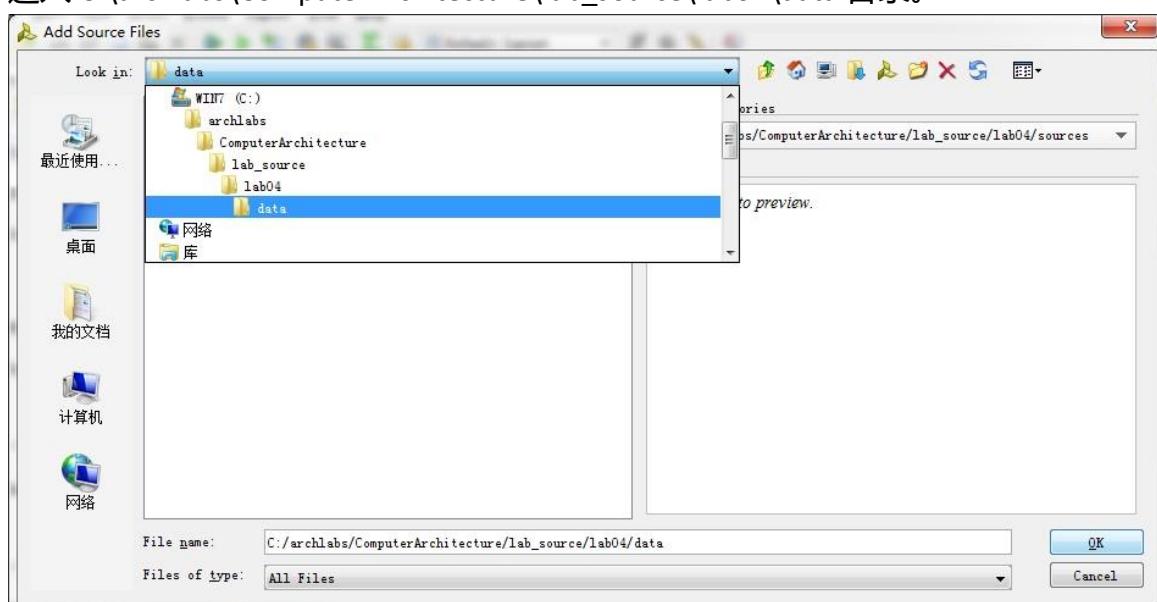


图 2-23 切换目录

选中全部 hex 文件，点击 OK。

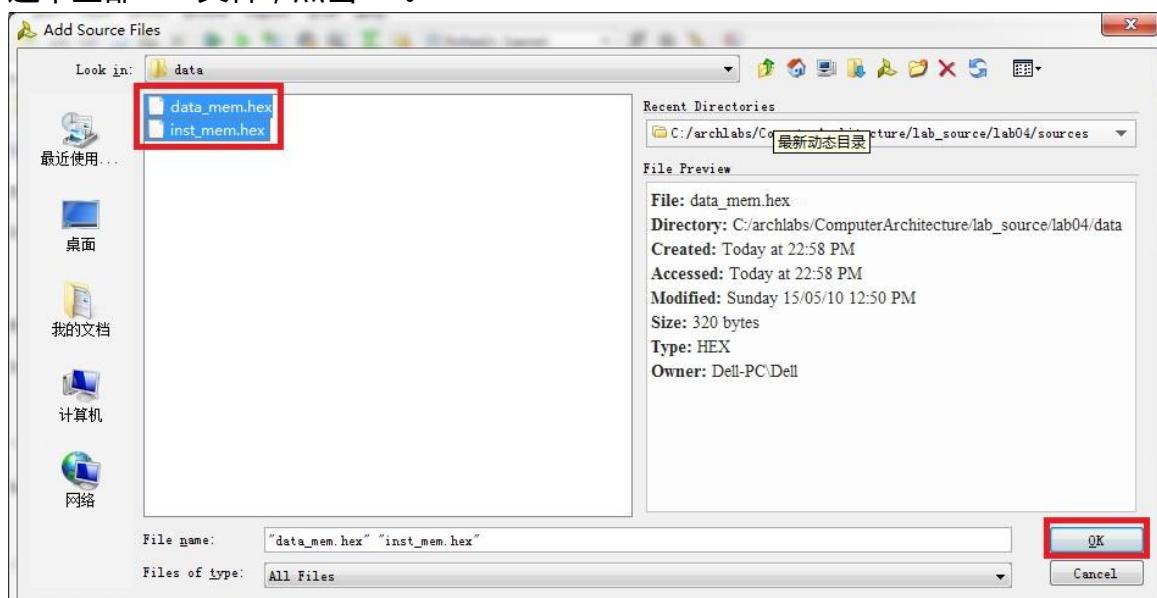
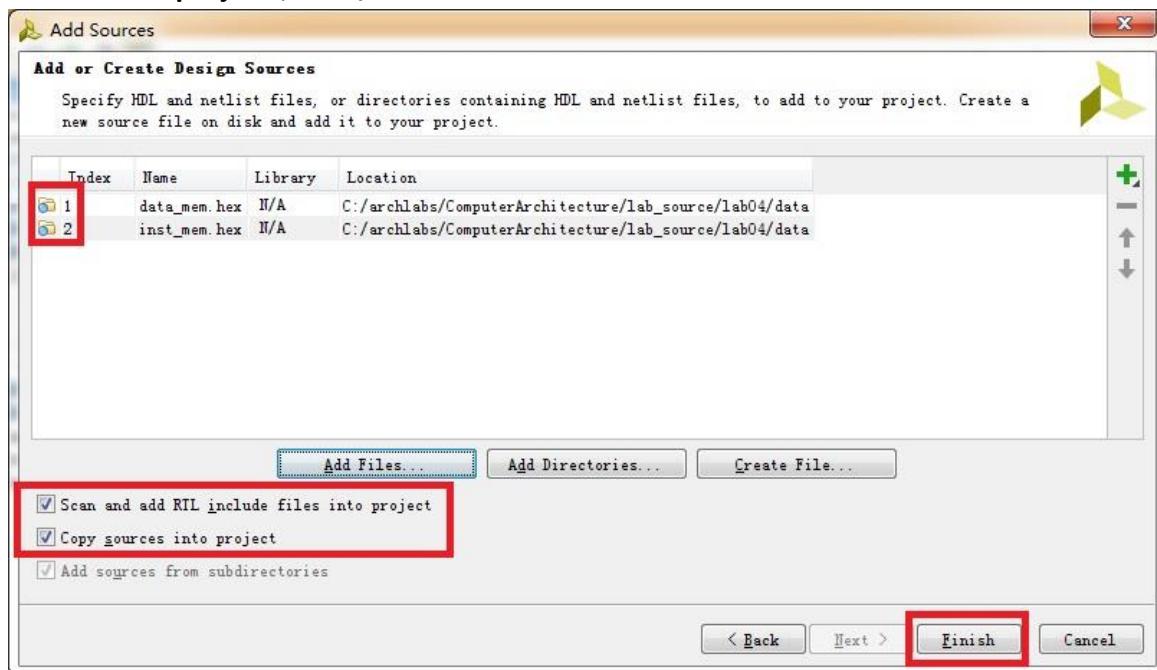


图 2-24 选中文件

标题	文档编号	版本 页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0 18 of 34
作者	修改日期	
Hawk	2019/1/19	公开

本次共加入文件 2 个。选中 Scan and add RTL include files into project 和 Copy sources into project , 点击 Finish。



点击 “+” 展开 Unknown , 用 Ctrl+鼠标左键选择 data_mem.hex 和 inst_mem.hex , 点击鼠标右键 , 点击 Set File Type。

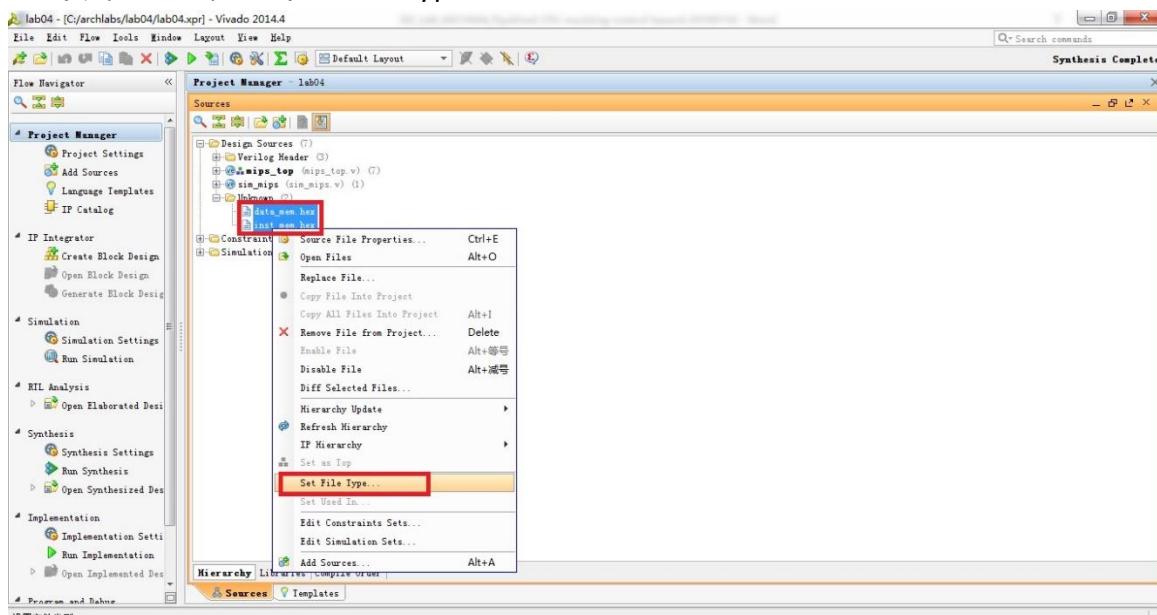


图 2-25 设置文件类型

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	19 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

设置 File type 为 Data Files , 点击 OK。

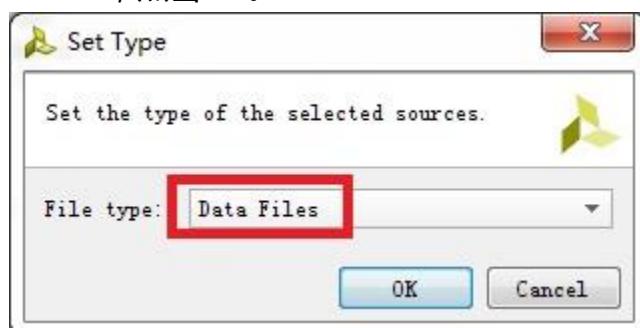


图 2-26 设置为 Data Files

在 Sources 窗口点击右键，点击 Refresh Hierarchy 更新视图。

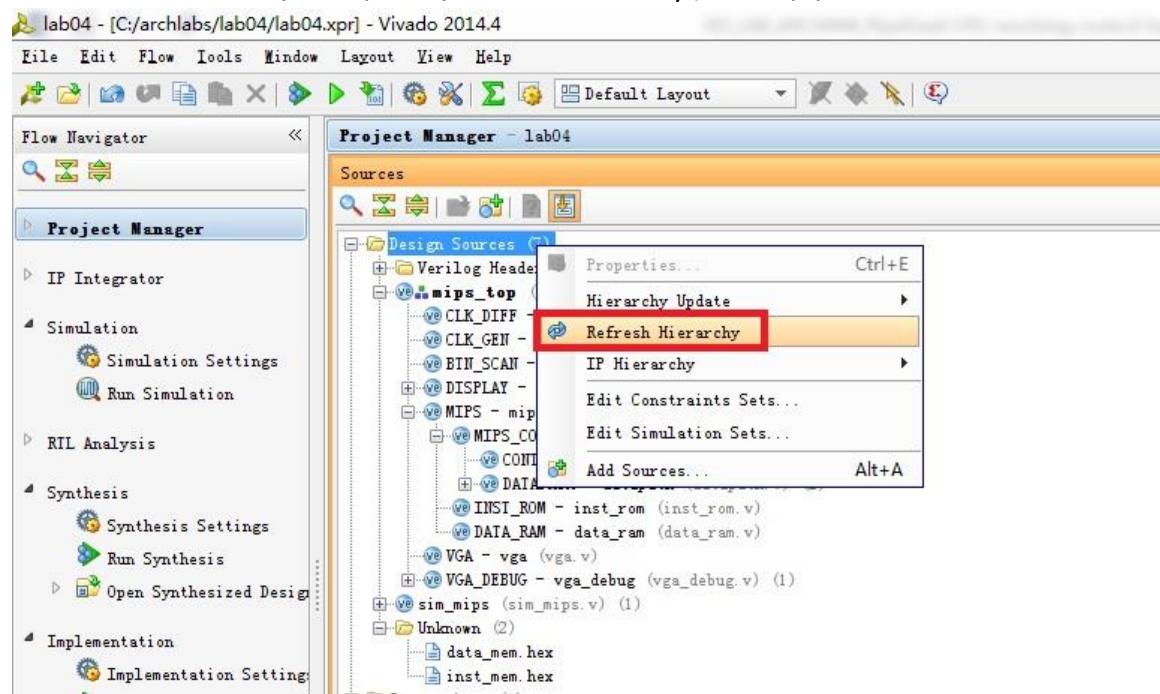


图 2-27 更新视图

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	20 of 34
作者 Hawk	修改日期 2019/1/19		公开

双击打开 mips_core.v 文件。

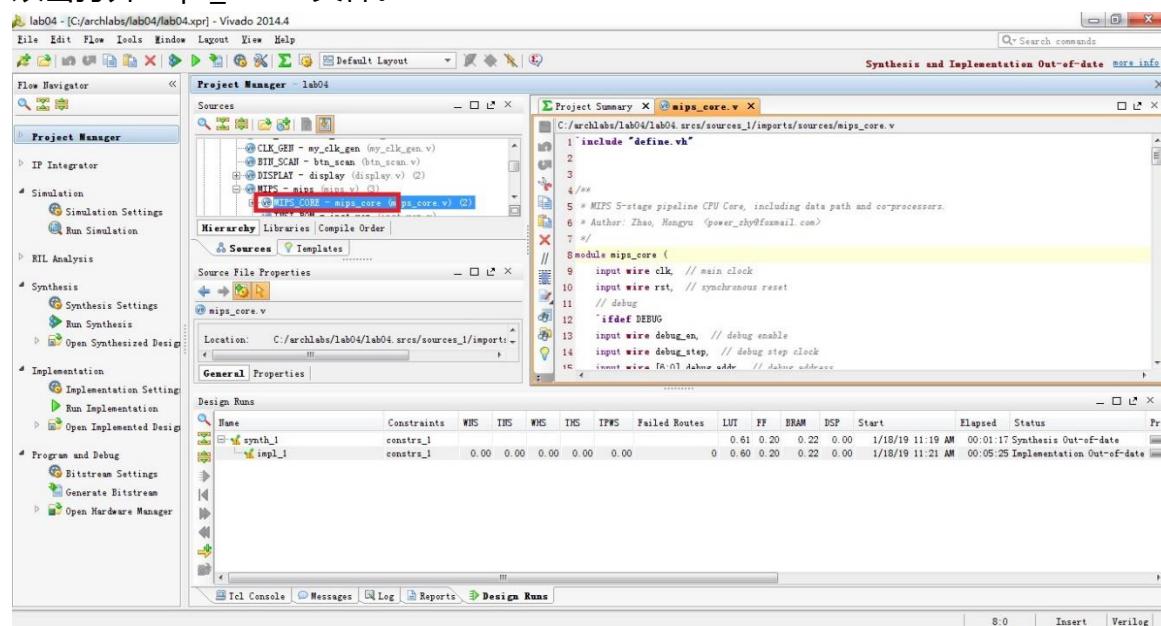


图 2-28 打开文件

在 mips_core.v 中添加 rs_rt_equal 和 fwd_m_ctrl 两个 wire 类型信号。

```

49
50   wire if_RST, if_EN, if_VALID;
51   wire id_RST, id_EN, id_VALID;
52   wire EXE_RST, EXE_EN, EXE_VALID;
53   wire MEM_RST, MEM_EN, MEM_VALID;
54   wire WB_RST, WB_EN, WB_VALID;
55
56   wire rs_rt_equal;
57   wire fwd_m_ctrl;
58
59 // controller
60 controller CONTROLLER (
61   .clk(clk),
62   .rst(rst),
63   //ifdef DEBUG

```

图 2-29 添加信号

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	21 of 34
作者 Hawk	修改日期 2019/1/19		公开

在 CONTROLLER 模块中，添加 rs_rt_equal 和 fwd_m 两个接口，删除 is_branch_exe 和 is_branch_mem。

```

controller CONTROLLER (
    .clk(clk),
    .rst(rst),
    `ifdef DEBUG
    .debug_en(debug_en),
    .debug_step(debug_step),
    `endif
    .inst(inst_data_ctrl),
    // .is_branch_exe(is_branch_exe),
    // .rs_rt_equal(rs_rt_equal),
    // .fwd_m(fwd_m_ctrl),
    .is_load_exe(is_load_exe),
    .regw_addr_exe(regw_addr_exe),
    .wb_wen_exe(wb_wen_exe),
    // .is_branch_mem(is_branch_mem),
);

```

图 2-30 修改代码

在 DATAPATH 模块中，添加 rs_rt_equal 和 fwd_m_ctrl 两个接口，删除 is_branch_exe 和 is_branch_mem。

```

datapath DATAPATH (
    .clk(clk),
    `ifdef DEBUG
    .debug_addr(debug_addr[5:0]),
    .debug_data(debug_data),
    `endif
    .inst_data_id(inst_data_ctrl),
    // .is_branch_exe(is_branch_exe),
    // .rs_rt_equal(rs_rt_equal),
    .fwd_m_ctrl(fwd_m_ctrl),
    .is_load_exe(is_load_exe),
    .regw_addr_exe(regw_addr_exe),
    .wb_wen_exe(wb_wen_exe),
    // .is_branch_mem(is_branch_mem),
    .is_load_mem(is_load_mem),
);

```

图 2-31 修改代码

标题	文档编号	版本	页	
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	22 of 34	
作者	修改日期			
Hawk	2019/1/19			公开

点击 File 菜单的 Save File 保存文件。

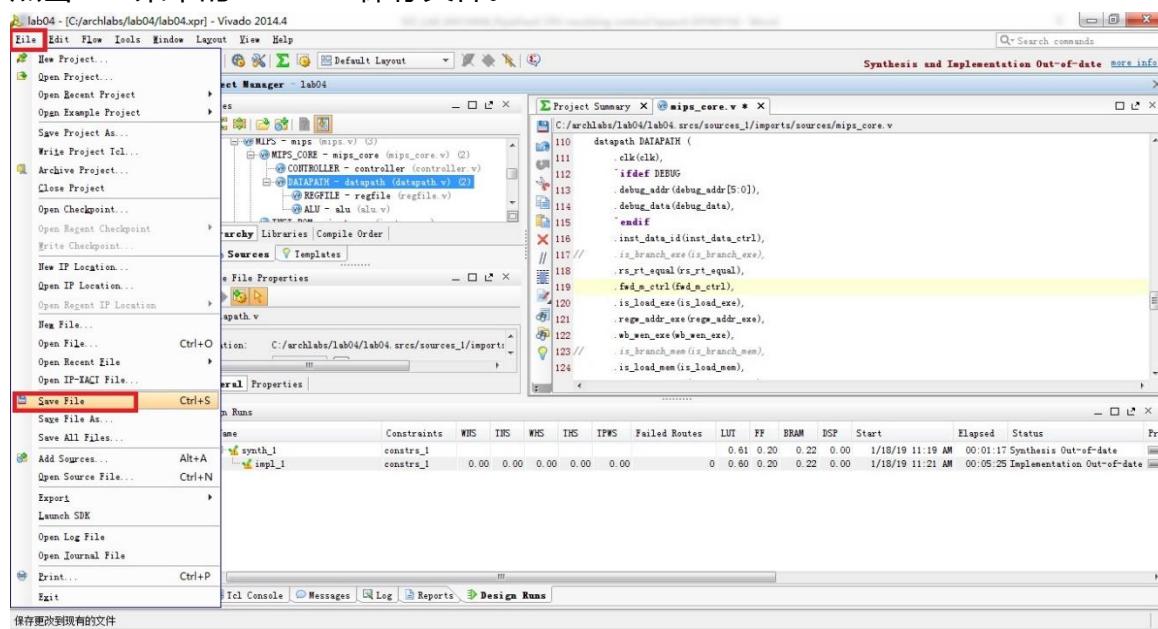


图 2-32 保存修改

展开 Verilog Header，打开 mips_define.vh 文件。加入参数 PC_BRANCH，删除参数 PC_BEQ 和 PC_BNE。

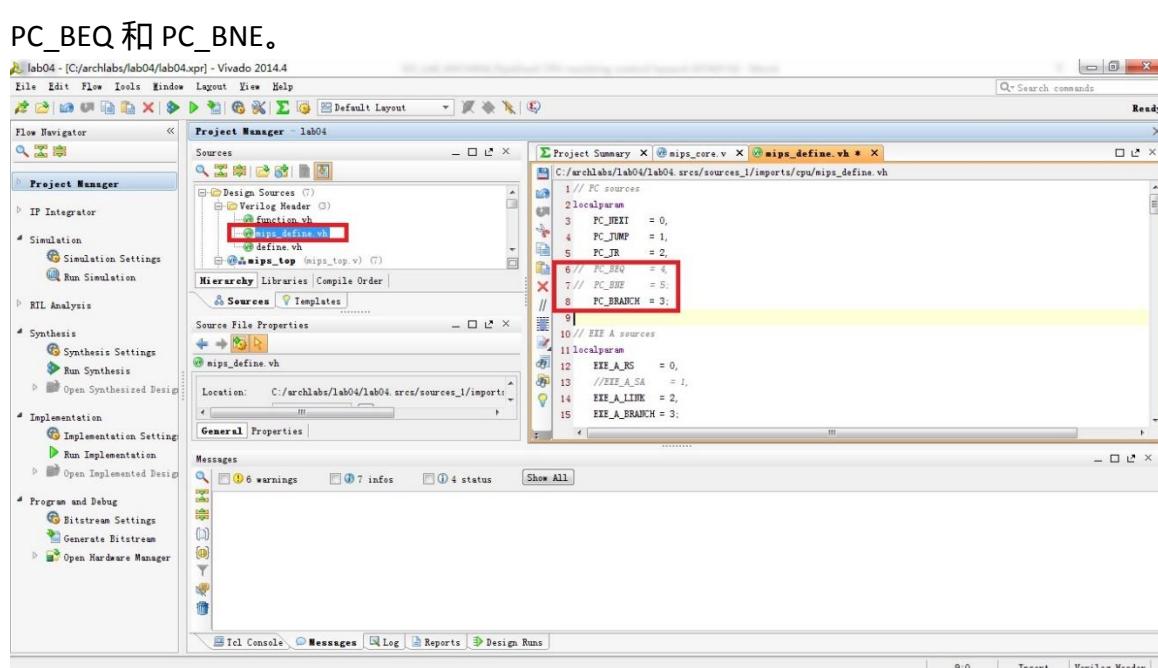


图 2-33 修改文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	23 of 34
作者 Hawk	修改日期 2019/1/19	公开	

点击 File 菜单的 Save File 保存修改。

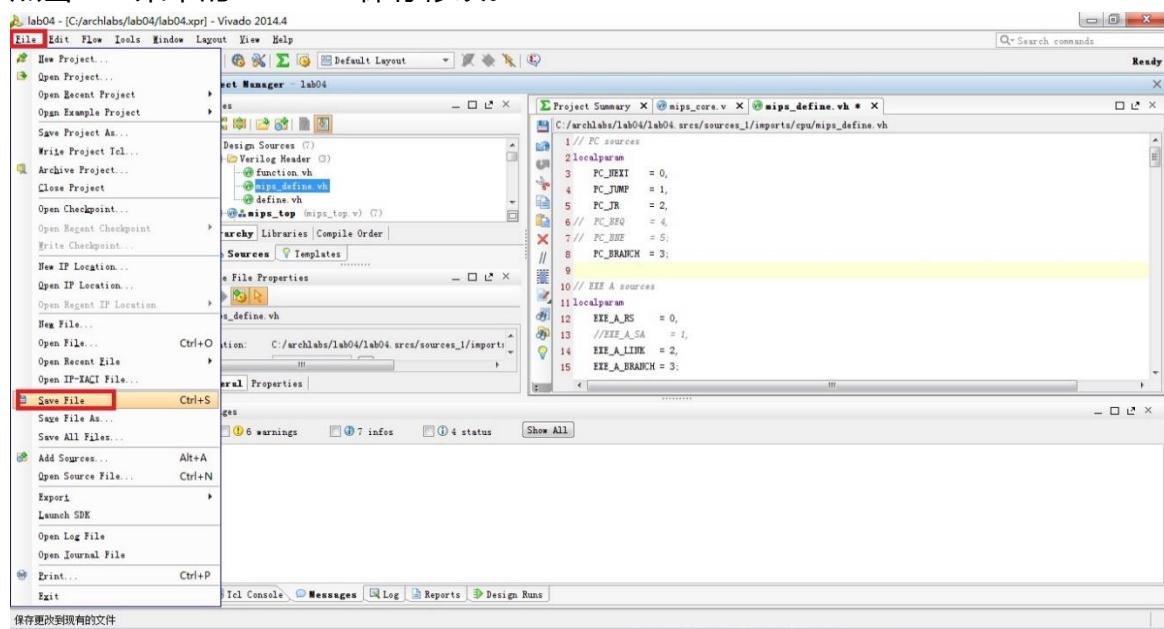


图 2-34 保存修改

2.3 功能仿真

进入 C:\archlabs\ComputerArchitecture\lab_source\lab04\simulation 目录，复制 sim_mips_behav.wcfg 文件。

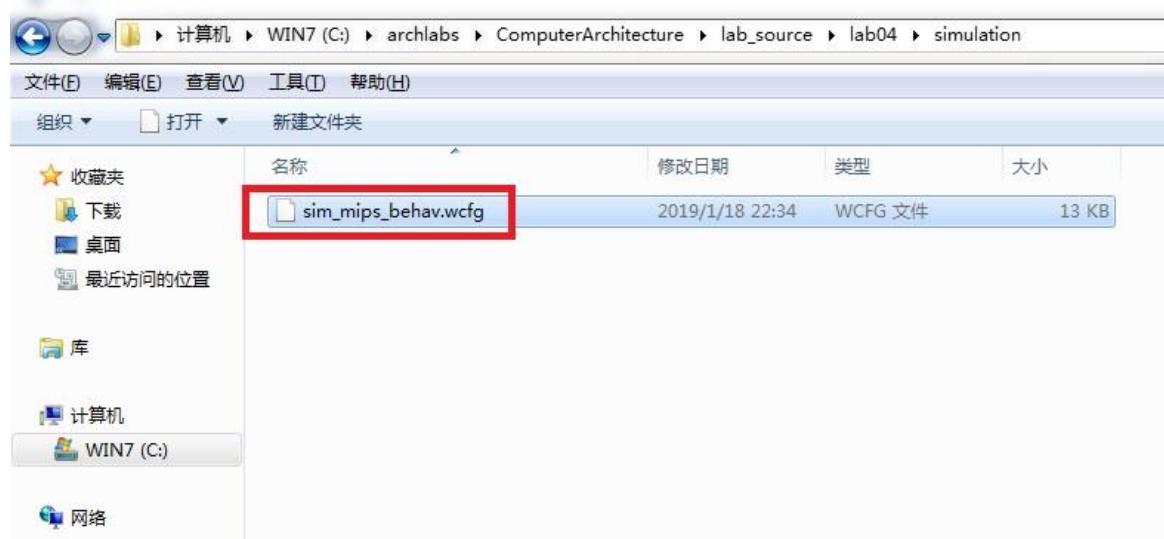


图 2-35 复制文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	24 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

将文件粘贴到当前 lab04 的工程目录下，这个文件包含了仿真的设置。

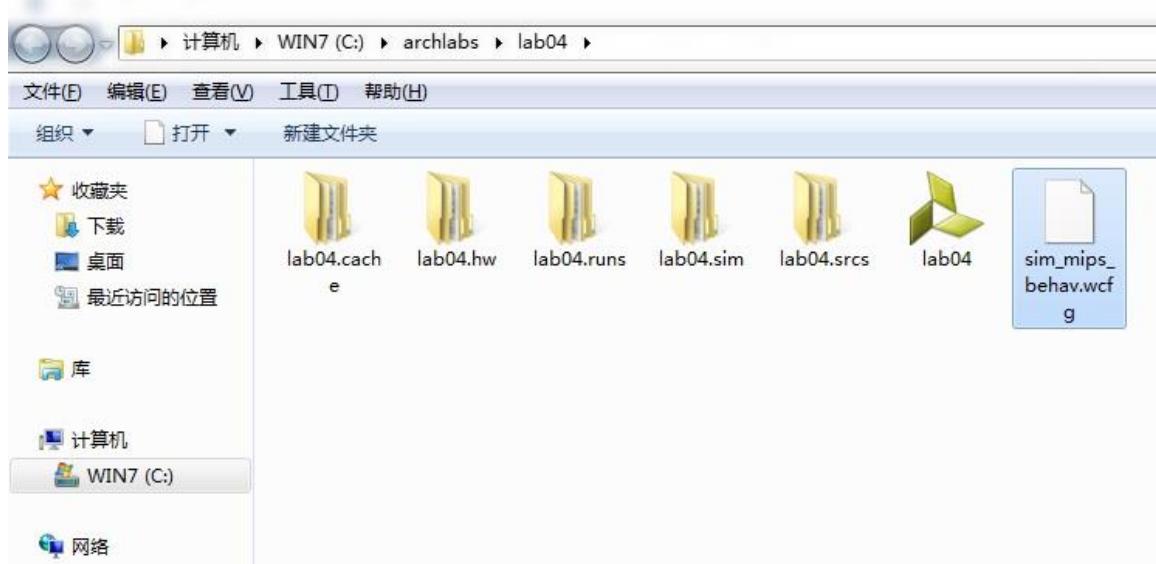


图 2-36 粘贴文件

点击 Run Simulation 下的 Run Behavioral Simulation。

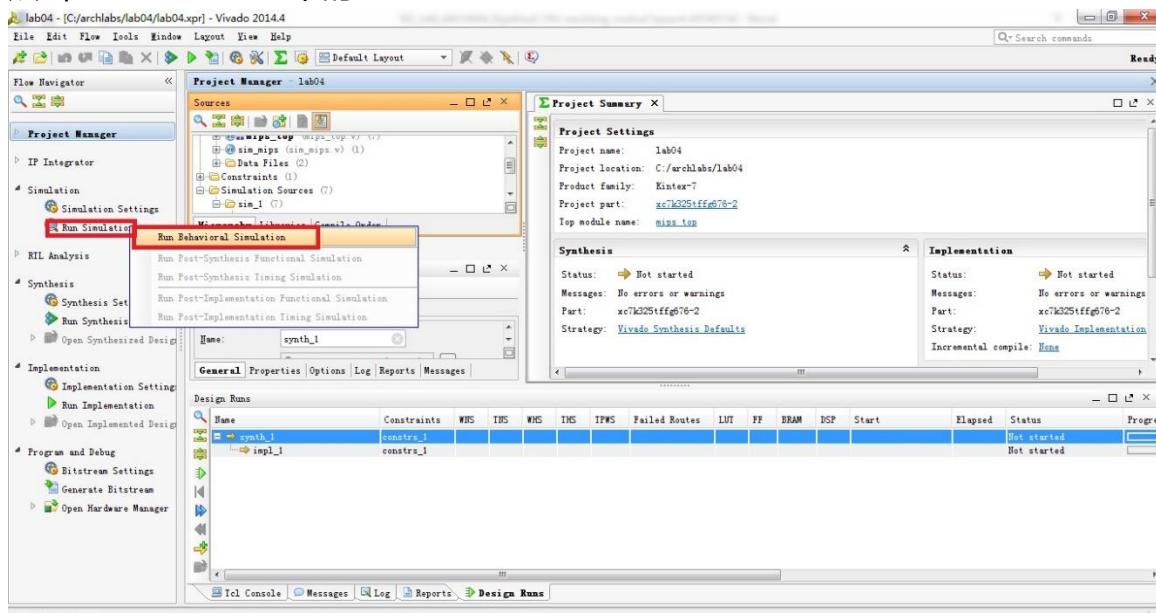


图 2-37 开始仿真

XINGDENG	标题	文档编号	版本	页
	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	25 of 34
作者	修改日期			
Hawk	2019/1/19			公开

仿真编译成功后，进入仿真视图。

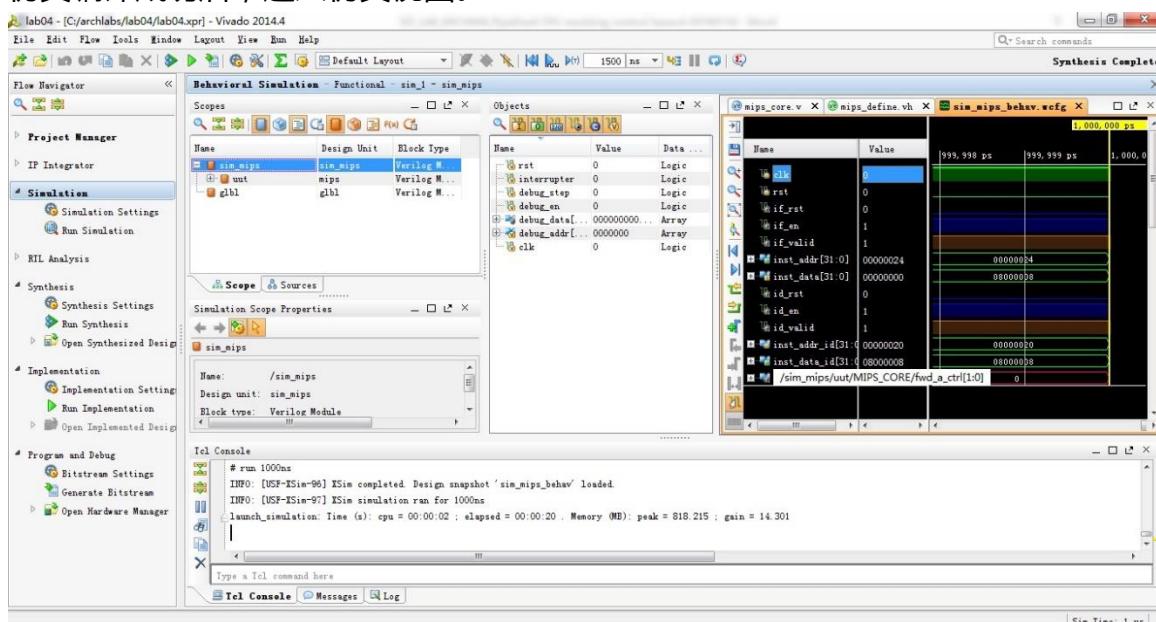


图 2-38 仿真视图

使用 Run for...进行仿真。

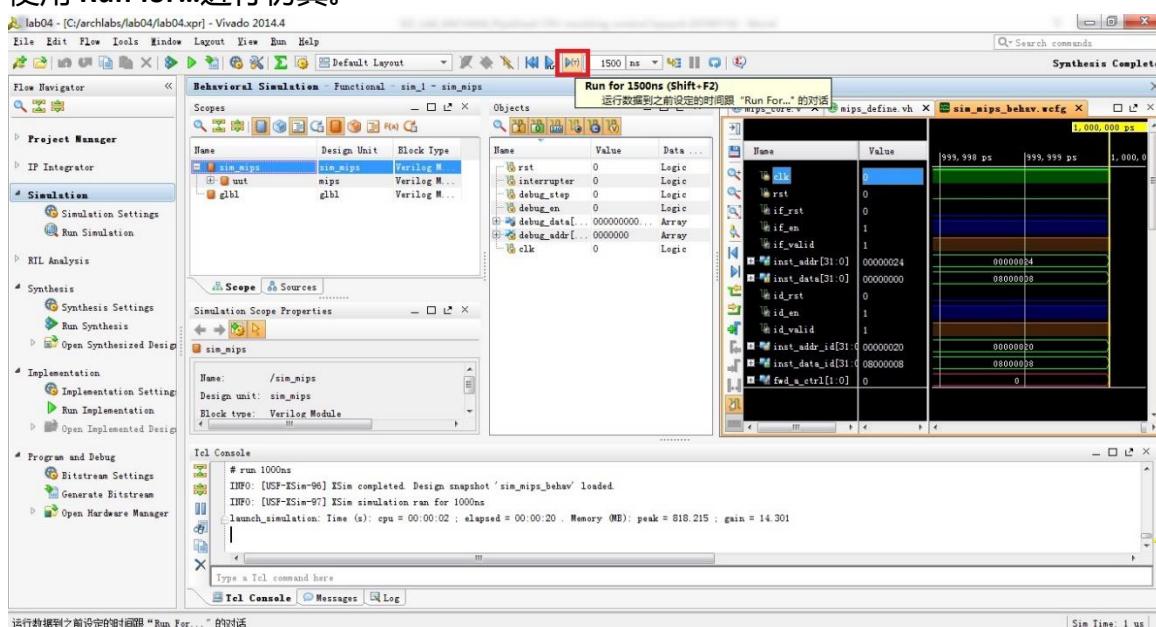


图 2-39 Run for...

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	26 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

通过键盘 Shift+ “-” 和 Shift+ “+” 可以对波形图进行缩放，将波形缩放到方便观察的程度。

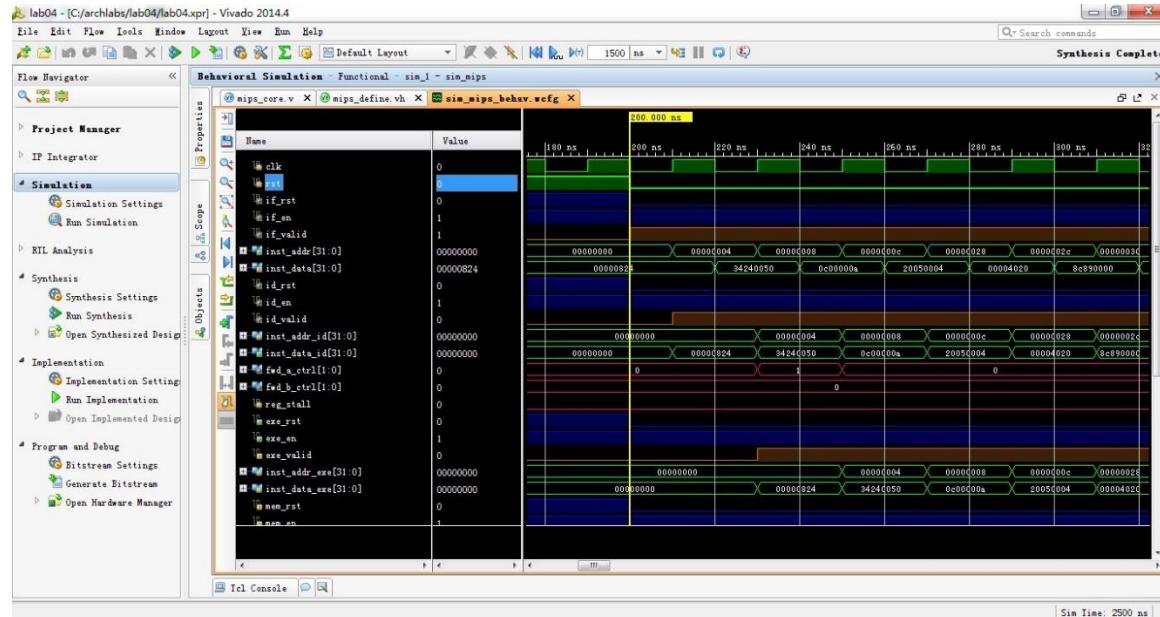


图 2-40 缩放波形

将仿真出的波形和本文档提供的波形对比，可以知道设计的电路是否正常工作。

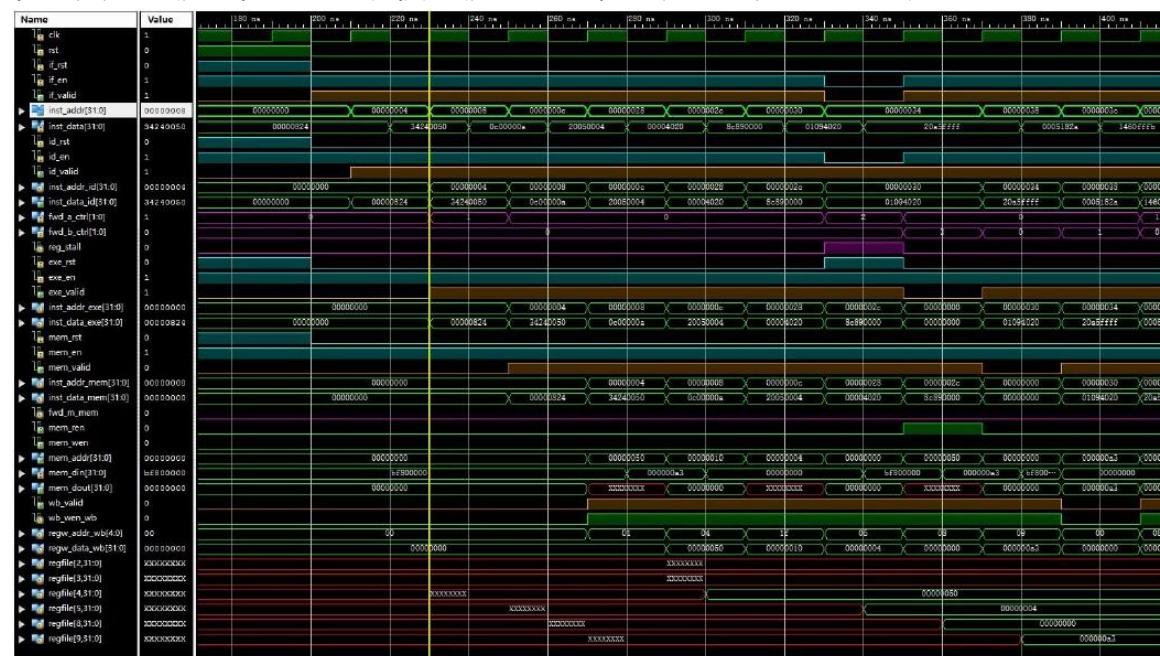


图 2-41 仿真结果 1

标题	文档编号	版本	页
XINGDENG	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0 27 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

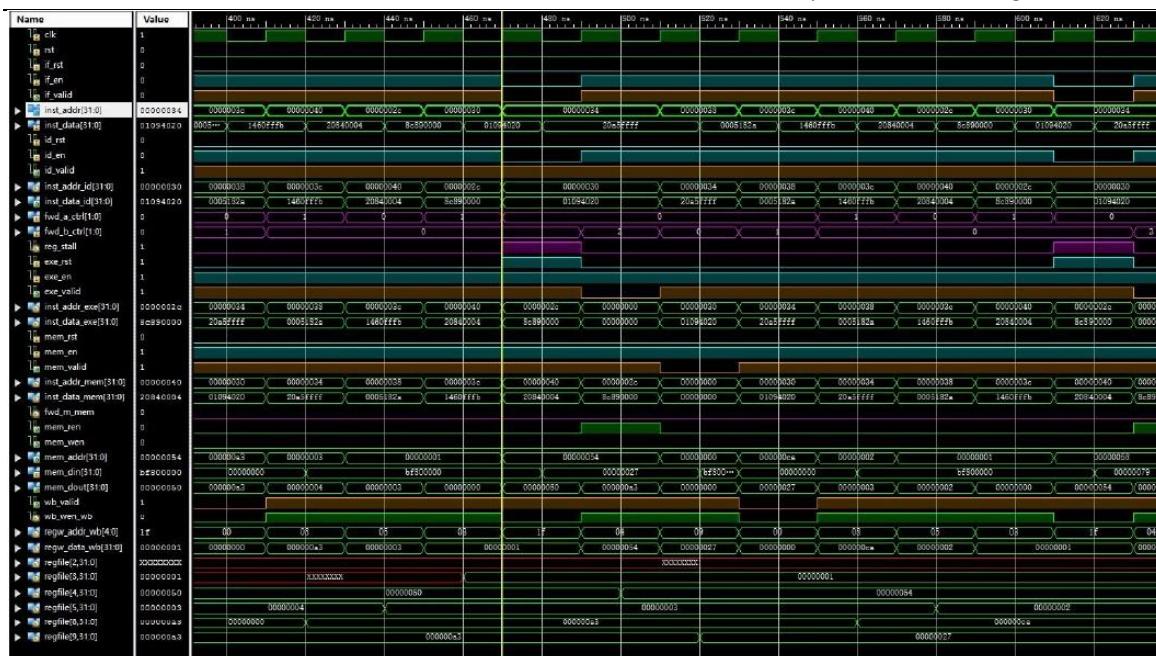


图 2-42 仿真结果 2

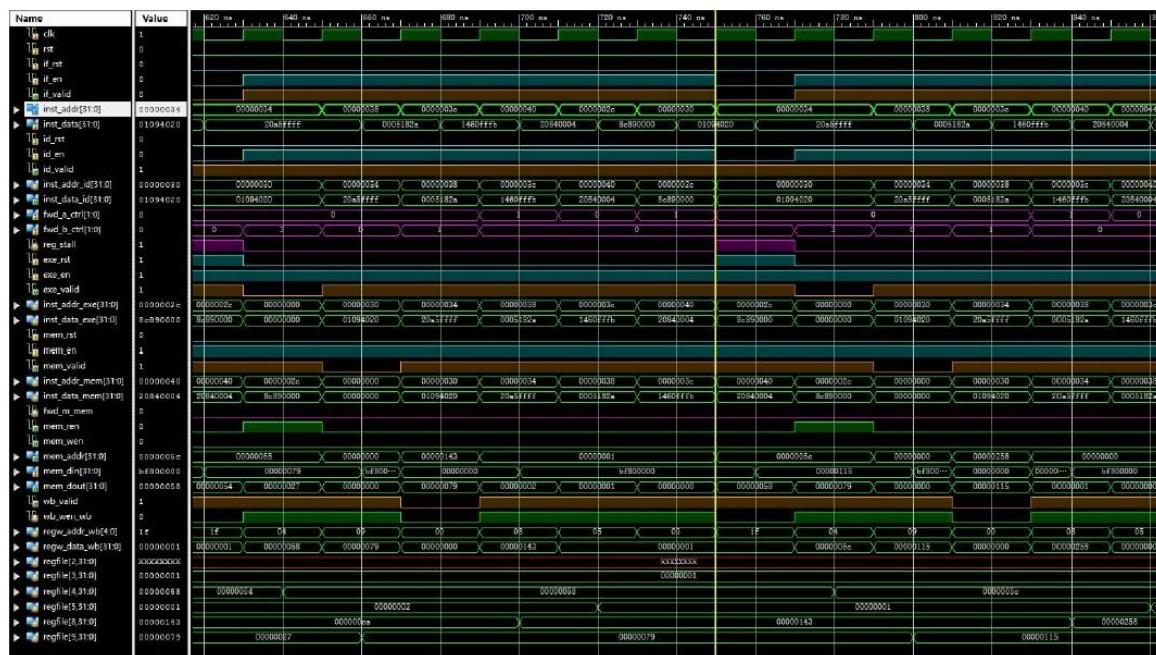


图 2-43 仿真结果 3

2.4 生成 bitstream 文件

标题	文档编号	版本	页
XINGDENG	Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0 28 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

右键点击 Simulation，点击 Close Simulation，结束仿真。

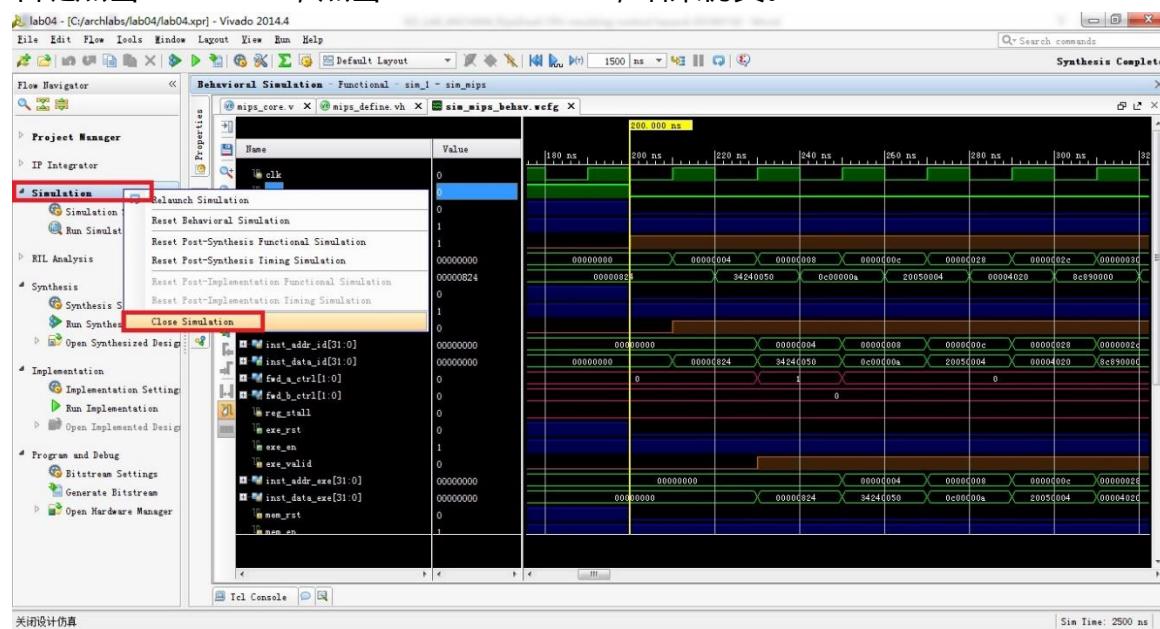


图 2-44 关闭仿真

mips_top.v 被设置为顶层文件后。点击 Generate Bitstream 生成 FPGA 配置文件。

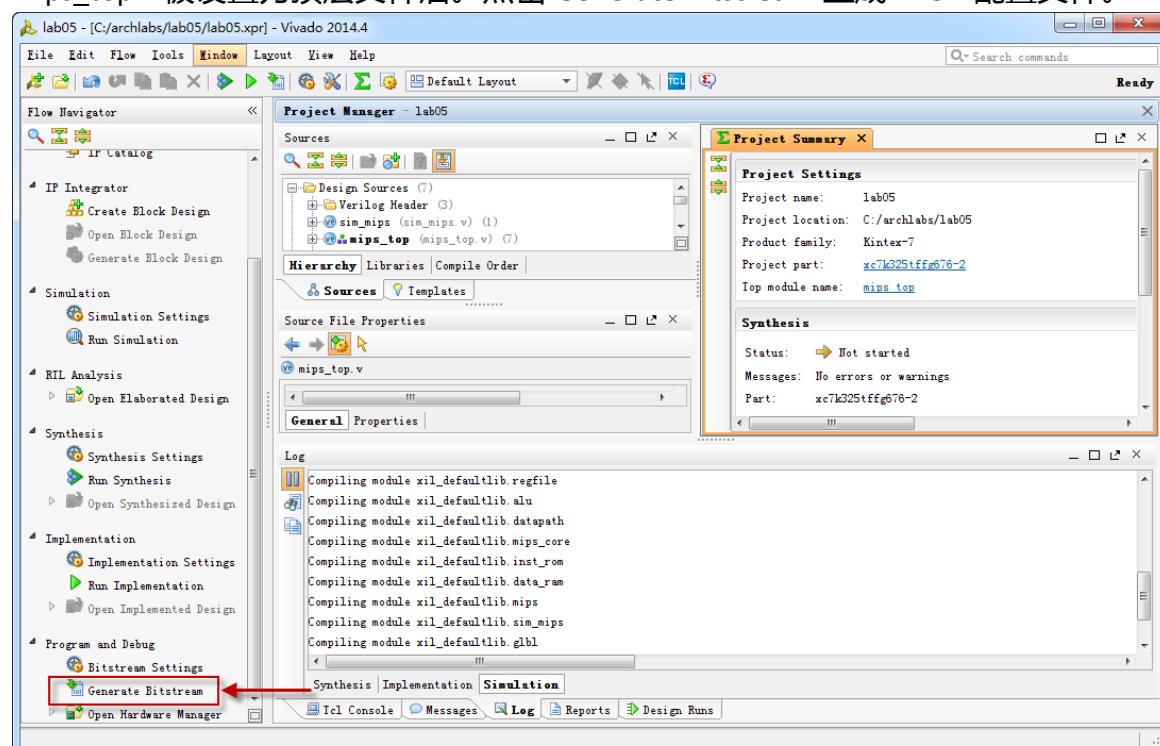


图 2-45 生成 bitstream

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	29 of 34
作者 Hawk	修改日期 2019/1/19		公开

在询问对话框中选择 Yes , 自动运行 synthesis 和 implementation。

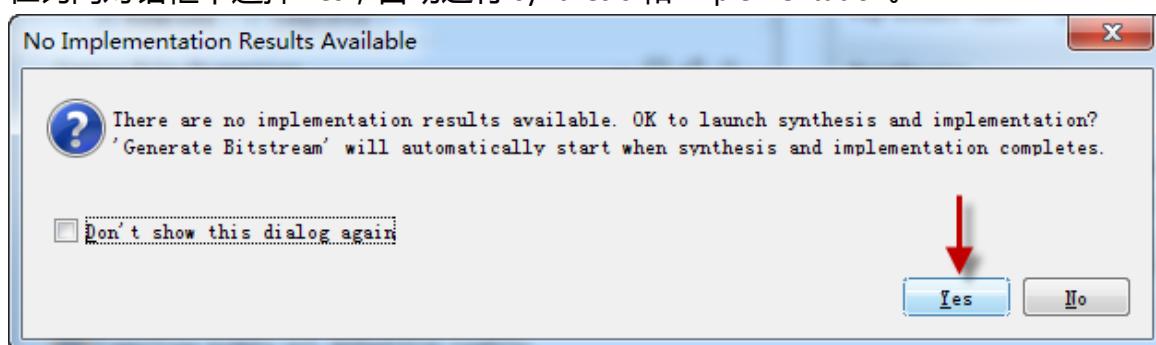


图 2-46 自动运行中间步骤

当显示 Bitstream Generation Completed 时表示文件成功生成 , 点击 Cancel 结束。

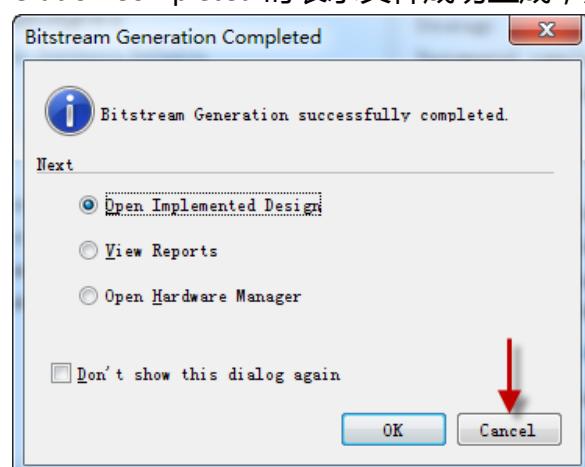


图 2-47 生成 bitstream 成功

2.5 下载验证

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	30 of 34
作者 Hawk	修改日期 2019/1/19		公开

仿真完成后，下载 bit 文件到 SWORD4.0 开发平台查看真实效果。按图连接 SWORD4.0 的 12V 电源和 JTAG，然后打开开关。

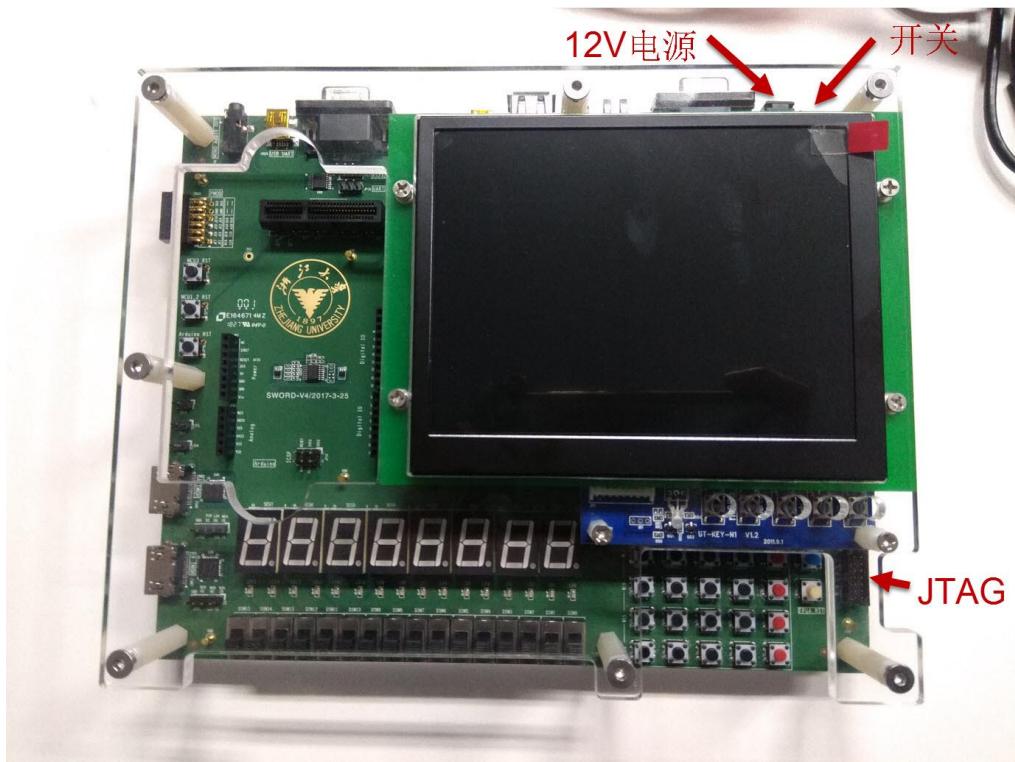


图 2-48 连接 SWORD4.0

SWORD4.0 的控制开关主要有拨码开关、键盘矩阵、复位按钮等。

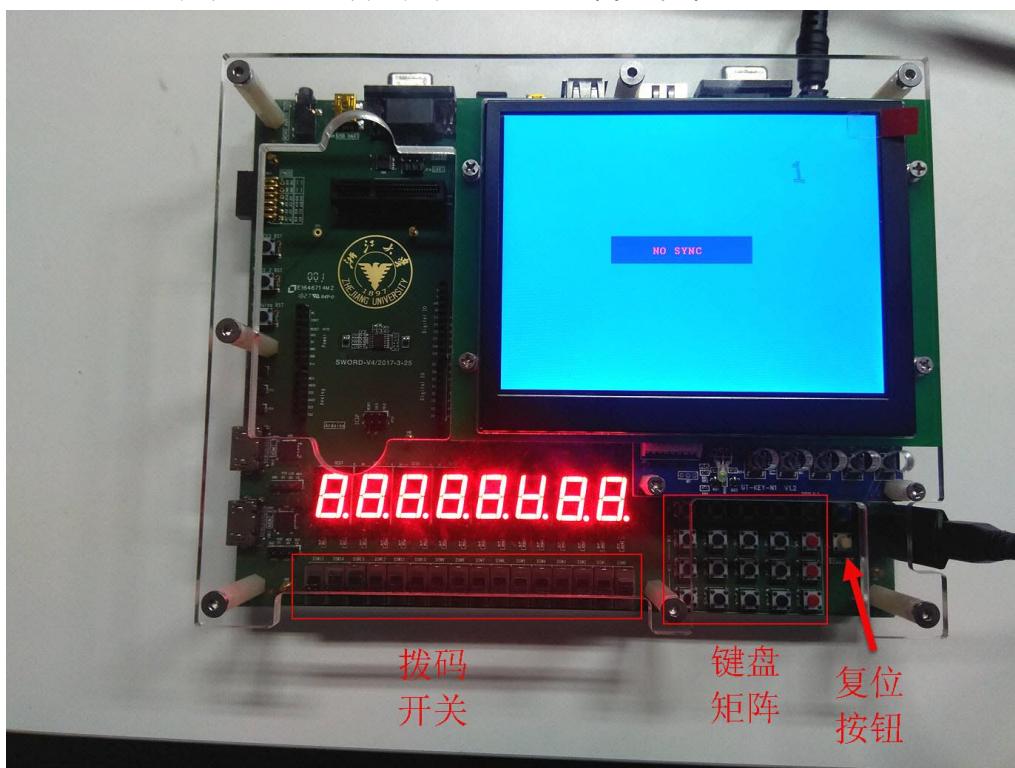


图 2-49 主要控制 I/O

标题	文档编号	版本	页
XINGDENG Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	31 of 34
作者 Hawk	修改日期 2019/1/19		公开

在 Vivado 中点击 Open Hardware Manager。

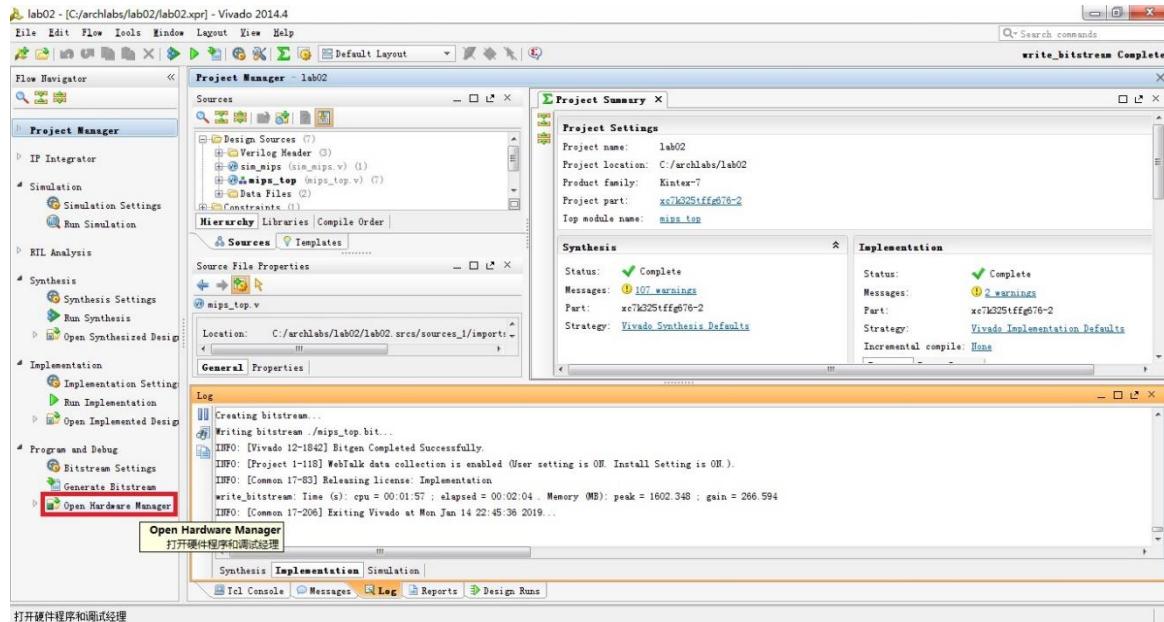


图 2-50 进入 Hardware Manager

点击 Open target，然后点击 Auto Connect。

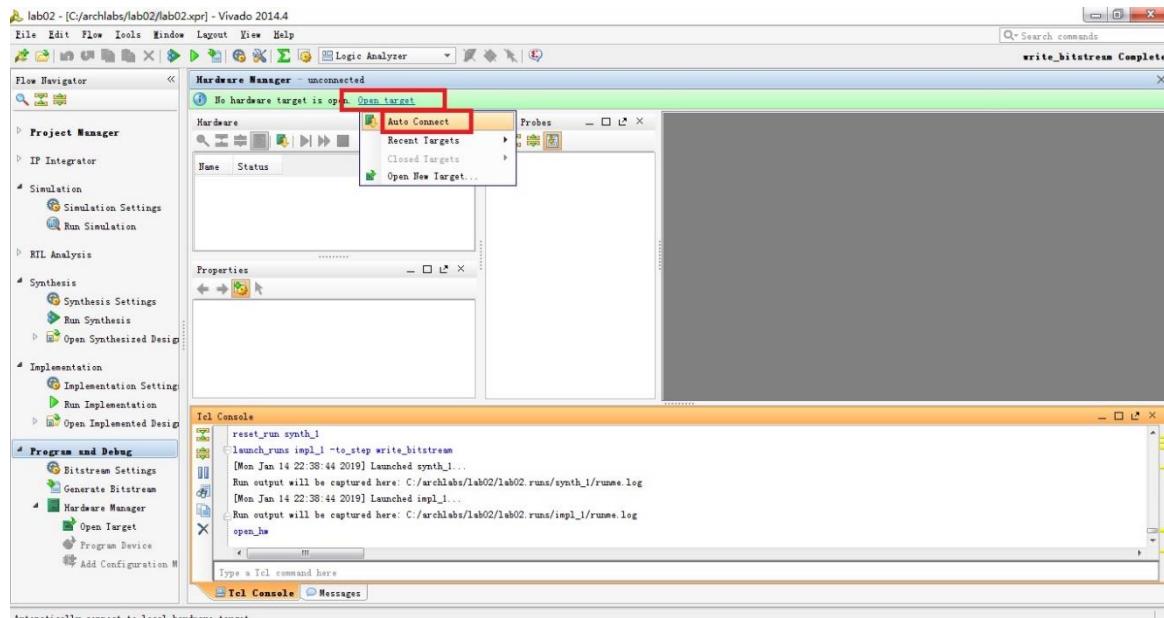


图 2-51 自动连接到 SWORD4.0

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	32 of 34
作者 Hawk	修改日期 2019/1/19	公开	

右键点击找到的 xc7k325t_0 , 选择 Program Device。

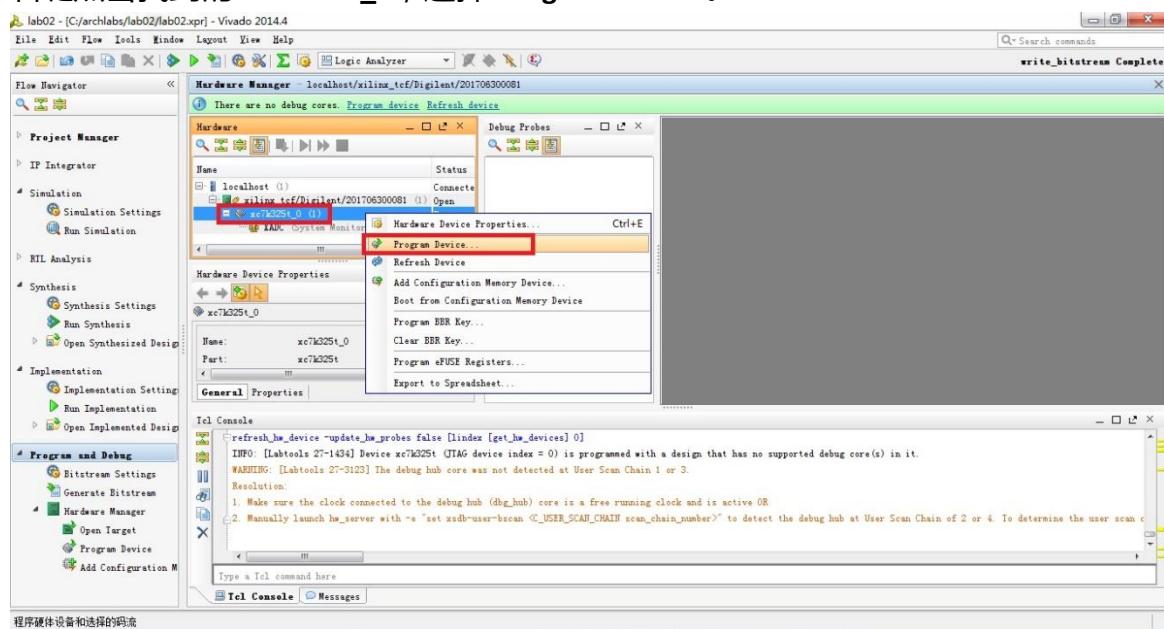


图 2-52 Program Device

点击 Program 下载 bit 文件。

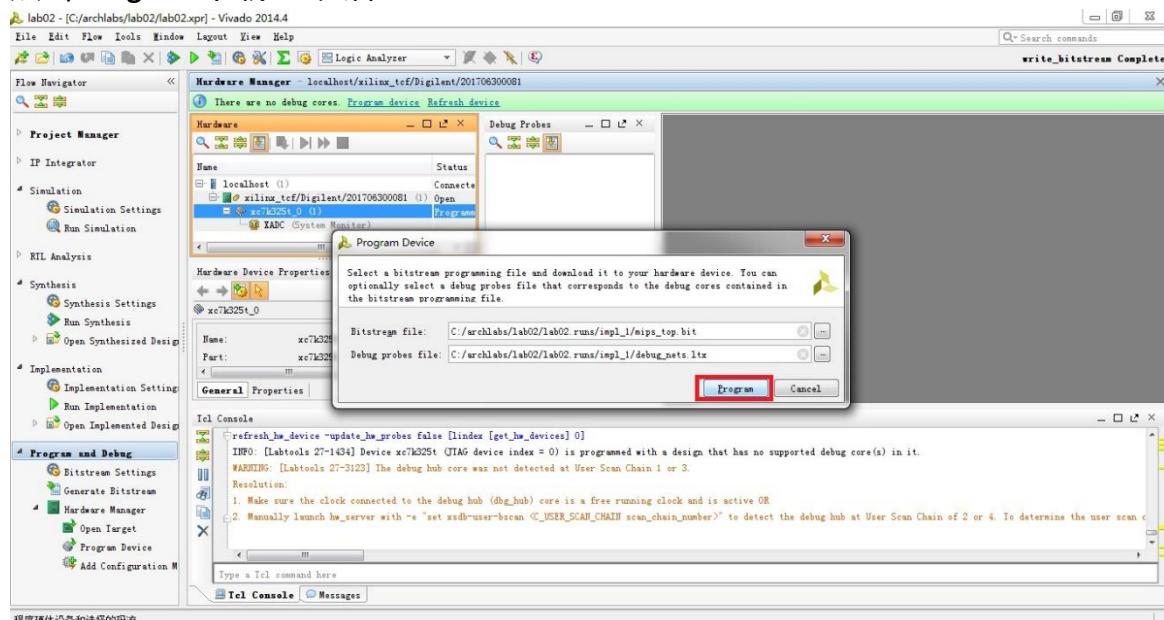


图 2-53 下载 bit 文件

标题	文档编号	版本	页
Lab4: Pipelined CPU resolving control hazard	XD-LAB-ARCH-004	1.0	33 of 34
作者	修改日期		
Hawk	2019/1/19	公开	

下载完成后 SWORD4.0 上的反应，液晶屏上会显示调试信息。

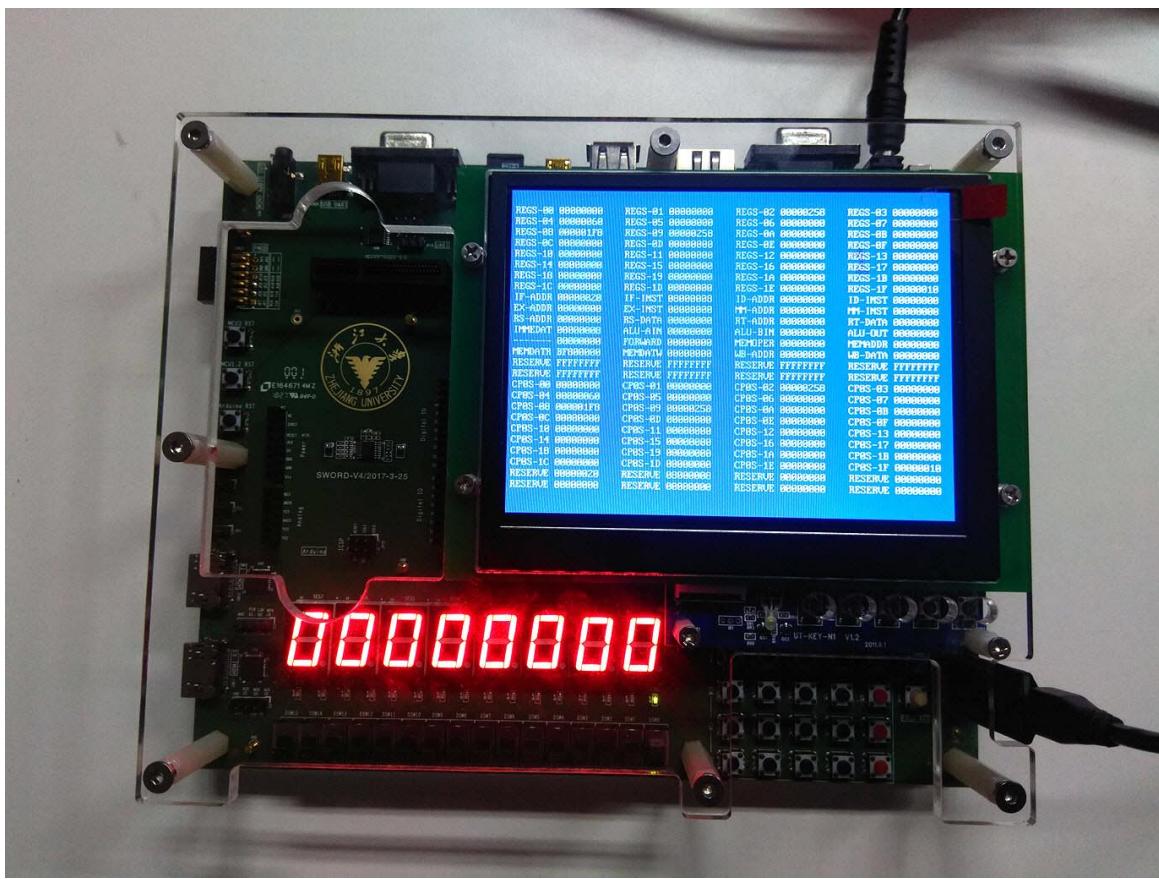


图 2-54 运行效果

通过将拨码开关 DSW0 拨向上，可以开启单步模式。之后每按一下键盘矩阵左下角的 BTNX4Y0，CPU 就执行一条指令。按复位键 FPGA RST 按钮回到初始状态。

在单步模式下，通过每一步液晶屏显示的数据与仿真波形的对比，查看流水线 cpu 是否正常工作。

标题	文档编号	版本	页	
XINGDENG	Lab4: Pipelined CPU resolving control hazard			
作者	修改日期			
Hawk	2019/1/19	公开		